TUGAS AKHIR – KI141502

Klasifikasi Kepribadian Berdasarkan Tulisan dari Twitter Menggunakan Metode Naive Bayes, KNN dan SVM

Bayu Yudha Pratama
NRP 5111100127

Dosen Pembimbing
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Ratih Nur Esti A., S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015

FINAL PROJECT – KI141502

Personality Classification Based on Twitter Text Using Naive Bayes, KNN and SVM

Bayu Yudha Pratama
NRP 5111100127

Advisor
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Ratih Nur Esti A., S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

LEMBAR PENGESAHAN

Klasifikasi Kepribadian Berdasarkan Tulisan dari Twitter Menggunakan Metode Naive Bayes, KNN dan SVM

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :
BAYU YUDHA PRATAMA
NRP : 5111 100 127

Disetujui oleh Dosen Pembimbing tugas akhir :

Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D. .....
NIP: 19590803 198601 1 001 (pembimbing 1)

Ratih Nur Esti A., S.Kom., M.Sc. .....
NIP: 19841210 201404 2 003 (pembimbing 2)

SURABAYA
JUNI 2015

Klasifikasi Kepribadian Berdasarkan Tulisan dari Twitter Menggunakan Metode Naive Bayes, KNN dan SVM

Nama Mahasiswa	: Bayu Yudha Pratama
NRP	: 5111100127
Jurusan	: Teknik Informatika FTIf-ITS
Dosen Pembimbing 1	: Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc.,
Dosen Pembimbing 2	: Ratih Nur Esti A., S.Kom., M.Sc.

ABSTRAK

Kepribadian merupakan komponen dasar dari perilaku manusia. Kepribadian telah terbukti memengaruhi interaksi dan preferensi seorang individu. Hingga saat ini, untuk mengukur kepribadian sesorang, mereka diharuskan mengambil tes kepribadian.

Media sosial adalah tempat dimana seorang individu mengekspresikan dirinya kepada dunia luar. Tulisan yang dibuat oleh pengguna media sosial dapat dianalisis untuk mendapatkan informasi yang diinginkan.

Tugas akhir ini memprediksi kepribadian berdasarkan teks yang dituliskan oleh pengguna media sosial Twitter. Bahasa yang digunakan adalah bahasa Indonesia dan bahasa Inggris. Metode klasifikasi yang diimplementasikan adalah Naive Bayes, K-Nearest Neighbors dan Support Vector Machine. Hasil uji coba menunjukkan metode Naive Bayes dengan rata-rata akurasi 63% sedikit mengungguli metode lainnya.

Kata Kunci**: **Identifikasi kepribadian, K-Nearest Neighbors, Klasifikasi teks, Machine learning, Media sosial, Naive Bayes, Support Vector Machine.

Personality Classification Based on Twitter Text Using Naive Bayes, KNN and SVM

Student Name	: Bayu Yudha Pratama
Student ID	: 5111100127
Major	: Teknik Informatika FTIf-ITS
Advisor 1	: Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc.,
Advisor 2	: Ratih Nur Esti A., S.Kom., M.Sc.

ABSTRACT

Personality is a fundamental basic of human behavior. Personality has been shown to affect the interaction and preferences of an individual. Until now, to gauge their personalities people are required to take a personality test.

Social media is a place where users expresses themselves to the world. Posts made by users of social media can be analyzed to obtain personal information.

This final project is to predict personality based on a text written by Twitter users. The language used is Indonesian and English. Classification method implemented is Naive Bayes, K-Nearest Neighbors and Support Vector Machine. Testing results showed Naive Bayes with an average accuracy of 63% slightly outperformed the other methods.

Keywords: K-Nearest Neighbors, Machine learning, Naive Bayes, Personality identification, Social media, Support Vector Machine, Text classification.

KATA PENGANTAR

Segala puji dan syukur ke hadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul: **"Klasifikasi Kepribadian Berdasarkan Tulisan dari Twitter Menggunakan Metode Naive Bayes, KNN dan SVM".**

Dalam pelaksanaan dan pembuatan tugas akhir ini tentunya penulis tidak dapat menyelesaikannya tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Orang tua dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan tugas akhir ini.
2. Bapak Riyanarto Sarno dan Ibu Ratih Nur Esti Anggraini selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Ibu Henning Titi Ciptaningtyas selaku dosen wali selama masa perkuliahan.
4. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah dengan sabar mendidik dan memberikan ilmu bagi penulis.
5. Seluruh staf dan karyawan Jurusan Teknik Informatika ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
6. Teman-teman angkatan 2011 jurusan Teknik Informatika ITS yang memberikan dorongan motivasi dan bantuan kepada penulis.
7. Serta pihak-pihak lain yang tidak dapat disebutkan satu per satu.

Harapan dari penulis semoga apa yang tertulis di dalam buku tugas akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan serta dapat memberikan kontribusi yang nyata pada masyarakat.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan. Penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Juni 2015

Bayu Yudha Pratama

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR KODE SUMBER

BAB I
PENDAHULUAN

Bab ini akan memaparkan garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Personality atau kepribadian adalah kombinasi dari sifat dan tingkah laku seseorang dalam menghadapi berbagai situasi. Kepribadian seseorang dapat memengaruhi pemilihan individu tersebut dalam berbagai hal seperti laman, buku, musik maupun film [1]. Selain itu kepribadian juga memengaruhi interaksi individu tersebut terhadap orang lain dan lingkungan sekitar. Kepribadian seseorang menjadi salah satu tolok ukur penilaian dalam berbagai bidang seperti seleksi dalam perekrutan pegawai, konseling karir, konseling hubungan/*relationship* maupun konseling kesehatan.

Selama ini, untuk mengetahui kepribadian seseorang, orang tersebut harus mengikuti berbagai tes kepribadian. Tes kepribadian dapat berupa deskripsi diri, wawancara, maupun observasi yang dilakukan ahli psikologi. Tentunya ini kurang praktis dan cukup merepotkan. Belakangan ini juga dikembangkan tes kepribadian dengan metode kuesioner yang dapat dilakukan pengguna di dunia maya [2], namun tetap saja cara ini kurang praktis karena pengguna masih harus mengisi beberapa pertanyaan. Ciri kepribadian seseorang dapat diperoleh secara otomatis dari teks yang dituliskannya [3]. Pemilihan kata-kata yang sering digunakan dapat menggambarkan kepribadian orang tersebut.

Situs media sosial adalah tempat dimana pengguna merepresentasikan dirinya terhadap dunia luar. Aktivitas yang dilakukan di media sosial seperti memberi komentar, tulisan dan pengubahan status dapat mengungkapkan informasi pribadi yang

dapat dimanfaatkan. Teks yang ditinggalkan oleh pengguna tersebut dapat dianalisis untuk mendapatkan informasi yang diinginkan.

Tugas akhir ini akan membahas tentang bagaimana memprediksikan kepribadian melalui teks yang dituliskan oleh pengguna media sosial Twitter. Metode Naive Bayes, K-Nearest Neighbors, dan Support Vector Machine digunakan untuk mengklasifikasikan teks kepada jenis kepribadian yang ada.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana mendapatkan data teks dari pengguna Twitter?
2. Bagaimana mengklasifikasikan data teks sesuai dengan model kepribadian?
3. Bagaimana membandingkan hasil yang diperoleh dari metode Naive Bayes, Support Vector Machine dan k-Nearest Neigbors?

1.3. Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Kumpulan kosa kata yang digunakan adalah bahasa Indonesia dan Inggris.
2. Klasifikasi data menggunakan metode *supervised learning* atau sudah dilabelkan sebelumnya.
3. Media sosial yang digunakan adalah Twitter.
4. Bahasa pemrograman yang digunakan adalah Python.

1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Memprediksi kepribadian pengguna media sosial dari tulisan yang dibuatnya.

2. Membandingkan hasil yang didapat dari metode yang digunakan.

1.5. Metodologi

Tahap yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. **Studi literatur**
   Pada tahap ini dilakukan tahap pengumpulan dan pembelajaran informasi yang akan digunakan untuk mengimplementasikan tugas akhir. Literatur yang digunakan adalah sebagai berikut: Big Five Personality Traits, Text Classification, Naive Bayes, K-Nearest Neighbors, Support Vector Machine, Scikit-Learn, NLTK dan kumpulan penelitian sebelumnya.

2. **Analisis dan Perancangan Sistem**
   Pada tahap ini dilakukan analisis, perancangan dan pendefinisian kebutuhan sistem untuk mengetahui permasalahan yang akan dihadapi pada tahap implementasi.

3. **Implementasi**
   Pada tahap ini dilakukan implementasi perangkat lunak berupa aplikasi berbasis web. Bahasa pemrograman yang digunakan adalah Python.

4. **Pengujian dan evaluasi**
   Pada tahap ini dilakukan pengujian dan evaluasi terhadap implementasi metode pada aplikasi.

5. **Penyusunan buku tugas akhir**
   Pada tahap ini dilakukan penyusunan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.6. Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, buku ini diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian sebagai berikut.

Bab 1. Pendahuluan
Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan dan sistematika penyusunan tugas akhir.

Bab II. Tinjauan Pustaka
Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan tugas akhir ini.

Bab III. Analisis dan Perancangan
Bab ini membahas mengenai analisis perangkat lunak meliputi analisis permasalahan, deskripsi umum perangkat lunak, perancangan sistem dan urutan pelaksanaan proses.

Bab IV. Implementasi
Bab ini berisi implementasi dari perancangan perangkat lunak.

Bab V. Pengujian dan Evaluasi
Bab ini membahas pengujian terhadap perangkat lunak yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi mengetahui kemampuan perangkat lunak.

Bab VI. Kesimpulan dan saran
Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka
Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

BAB II
TINJAUAN PUSTAKA

Bab ini memaparkan teori-teori yang menjadi dasar dari pembuatan tugas akhir. Teori-teori tersebut adalah sebagai berikut.

2.1. Big Five Personality Traits

Personality atau kepribadian adalah kombinasi dari sifat dan tingkah laku seseorang dalam menghadapi berbagai situasi. Kepribadian seseorang dapat mempengaruhi pemilihan individu tersebut dalam berbagai hal seperti laman web, buku, musik dan film [1]. Selain itu kepribadian juga mempengaruhi interaksi individu tersebut terhadap orang lain dan lingkungan sekitar. Kepribadian seseorang menjadi salah satu tolak ukur penilaian dalam berbagai bidang seperti seleksi dalam perekrutan pegawai, konseling karir, konseling hubungan maupun konseling kesehatan dan keselamatan. Kepribadian dibagi menjadi 5 kategori utama yang disebut Big Five Personality Model [4] yaitu:

- *Agreeableness* atau keramahan, berkaitan dengan fokus seseorang memelihara hubungan sosial dengan orang lain. *Agreeableness* tinggi cenderung mempercayai orang lain dan dapat berkompromi.
- *Conscientiousness* atau sifat kehati-hatian, berhubungan dengan organisasi kehidupan seseorang. Individu dengan *Conscientiousness* tinggi biasanya hidup teratur, dapat diandalkan dan konsisten. Sebaliknya individu dengan nilai rendah biasanya santai, spontan, kreatif dan toleran.
- *Extraversion* mengukur kecenderungan seseorang untuk mencari hubungan dan mengekspresikan dirinya terhadap orang lain. Seorang *Extrovert* cenderung ramah, aktif, energetik dan suka berbicara. Sebaliknya, *Introvert* lebih menghindari hubungan dengan orang lain.

- *Neuroticism* mengukur tingkat tendensi perubahan *mood*/suasana hati seseorang. Orang dengan *Neuroticism* tinggi berarti lebih mudah mengalami perubahan suasana hati dan terpengaruh dengan emosi negatif seperti *stress* atau gugup.
- *Openness* atau keterbukaan, berkaitan dengan imajinasi, kreativitas, keingintahuan dan apresiasi terhadap hal baru. Orang yang memiliki *Openness* tinggi memiliki sifat ingin tahu, menyukai perubahan, mengapresiasi sesuatu yang baru dan tidak biasa.

Big Five adalah model kepribadian yang paling banyak diteliti pada bidang psikologi. Big Five menunjukkan konsistensi pada wawancara, deskripsi diri dan observasi. Selain itu, Big Five juga konsisten ditemukan dalam berbagai usia dan budaya yang berbeda. Pengukuran Big Five banyak dilakukan melalui pengisian deskripsi diri maupun kuesioner [2].

2.2. Text Classification

Klasifikasi teks berarti menentukan suatu dokumen berupa teks ke dalam suatu kelas atau kategori [5, p1]. Sebelum memulai proses klasifikasi, data berupa teks harus terlebih dahulu diolah *(preprocessing*). Langkah *preprocessing* meliputi tokenisasi, *stemming* dan pembobotan.

- Tokenisasi adalah pemotongan kalimat berdasarkan tiap kata yang menyusunnya. Tokenisasi memecah kalimat menjadi kumpulan kata. Contoh: ‘saya akan pergi’ menjadi ‘saya’, ‘ akan’, ‘pergi’
- *Stemming* adalah proses yang mentransformasikan kata-kata yang terdapat pada suatu dokumen menjadi kata dasarnya (*root word*). *Stemming* dilakukan dengan menghilangkan imbuhan pada kata seperti ‘-nya, ‘-lah’, ‘di-‘.
- Pembobotan (*weighting*) dilakukan untuk membantu perhitungan. Salah satu metode pembobotan pada

klasifikasi teks adalah TF-IDF. *Term Frequency* (TF) adalah jumlah kemunculan kata pada suatu dokumen. *Document frequency* (DF) adalah jumlah dokumen dimana terdapat kata tersebut. Pembobotan TF-IDF adalah hasil perkalian nilai TF dan nilai *inverse* dari DF (ditunjukkan pada Persamaan 2.1).

$$tfidf_t = f_{t,d} \times log \frac{N}{df_t} \quad (2.1)$$

Keterangan:
$tfidf_t$ = bobot kata *t*
$f_{t,d}$ = jumlah kemunculan kata *t* pada dokumen *d*
N = jumlah total dokumen
df_t = jumlah dokumen dimana terdapat kata *t*

2.3. Naive Bayes

Naive Bayes adalah metode klasifikasi data berdasarkan penerapan teorema Bayes [6, p258-263]. Multinomial Naive Bayes (MNB) adalah variasi dari Naive Bayes yang didesain untuk menyelesaikan permasalahan klasifikasi dokumen teks. MNB memanfaatkan distribusi multinomial dengan jumlah kemunculan kata atau bobot kata sebagai fitur klasifikasi. Persamaan MNB ditunjukkan pada Persamaan 2.2**.**

$$c_{map} = \underset{c \in C}{\arg\max} \left[\log P(c) + \sum_{1 \le k \le n_d} \log P(t_k|c) \right] \quad (2.2)$$

Keterangan:
c_{map} = kelas pilihan
C = kumpulan semua kelas
n_d = jumlah kata pada dokumen *d*
$P(c)$ = probabilitas kelas *c*
$P(t_k|c)$ = probabilitas kata *t* ke-*k* pada kelas *c*

2.4. K-Nearest Neighbors

K-Nearest Neighbors (KNN) adalah algoritma klasifikasi yang menggunakan fungsi jarak antara data percobaan dengan data pelatihan serta jumlah tetangga terdekat untuk menentukan hasil klasifikasi [6, p297-301]. Fungsi jarak yang digunakan pada tugas akhir ini adalah *cosine similarity*. *Cosine similarity* adalah salah satu fungsi yang banyak digunakan dalam klasifikasi dokumen untuk menentukan kesamaan antara beberapa dokumen. Jarak yang dekat menunjukkan kesamaan antara 2 dokumen sehingga memiliki kategori yang sama. Persamaan *cosine similarity* yang digunakan dalam penilaian KNN ini ditunjukkan pada Persamaan 2.3. Penentuan kelas dilakukan dengan *voting* pada *K* tetangga yang terdekat. Tetangga terdekat merupakan *K* dokumen dengan nilai *similarity* tertinggi.

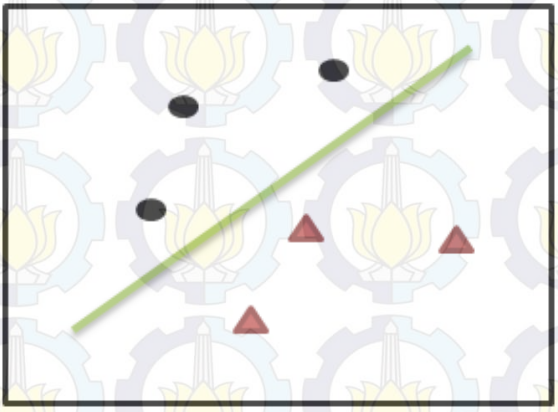
$$score(c, d_1) = \sum_{d_2 \in S_k d_1} I_c(d_2) \cos\ (vd_2, vd_1) \qquad (2.3)$$

Keterangan:
$score(c, d_1)$ = nilai skor dokumen uji pada kelas *c*
d_1 = dokumen uji
d_2 = dokumen latih
vd_1 = vektor dokumen uji
vd_2 = vektor dokumen latih
I_c = 1 jika d_2 adalah anggota kelas *c*, 0 jika tidak
$S_k d_1$ = kumpulan dokumen *k*-terdekat dari dokumen uji

2.5. Support Vector Machine

Support Vector Machine (SVM) adalah metode *supervised learning* yang menganalisis data dan mengenali pola yang digunakan untuk klasifikasi [6, p319-333]. SVM mengambil himpunan pelatihan data dan menandai sebagai bagian dari suatu kategori kemudian memprediksi suatu masukan apakah merupakan anggota dari kelas yang ada. Model SVM merepresentasi data sebagai titik dalam ruang, dipetakan sehingga

terpisah berdasarkan kategori yang dibagi oleh *hyperplane*/garis pemisah (Gambar 2.1).



Gambar 2.1 Contoh *hyperplane* pembagi kelas

Pada tahap klasifikasi, data percobaan yang dimasukkan kemudian dipetakan ke dalam ruang yang sama dan diperkirakan termasuk kategori berdasarkan sisi dan jarak data tersebut berada. Fungsi keputusan SVM ditunjukkan pada Persamaan 2.4.

$$f(x) = \text{sign}(w^T x + b) \quad (2.4)$$

Keterangan:
$f(x)$ = fungsi keputusan/kelas dokumen uji
w = *weight vector*
x = vektor dokumen uji
b = nilai *bias*

2.6. Twitter API

Twitter API (*Application Programming Interface*) adalah sekumpulan perintah, fungsi, komponen dan protokol yang disediakan oleh Twitter untuk mengambil atau memodifikasi data dari Twitter [7]. Twitter API dapat digunakan untuk membangun

aplikasi perangkat lunak, laman web, *widge*t dan beberapa proyek lain yang berinteraksi dengan Twitter. Fitur yang ditawarkan adalah modifikasi (*get* dan *post)* berbagai data dari pengguna Twitter seperti status, jaringan, media dan pengaturan.

Pengambilan data dari Twitter API memerlukan kode token *Application key* dan *Application secret* yang bisa didapatkan dengan mendaftarkan aplikasi yang dibuat di situs Twitter Developers. Terdapat batasan penggunaan pada API ini dalam suatu satuan waktu. Sebagai contoh, untuk sebuah token akses hanya diperbolehkan mengunakan 180 kueri *get* dalam rentang waktu 15 menit. Fungsi yang dipakai di tugas akhir ini adalah pengambilan status pengguna.

2.7. Scikit-Learn

Scikit-Learn adalah sebuah *open source library* yang mencakup *machine learning* untuk bahasa pemrograman Python [8]. Scikit-Learn menyediakan berbagai *learning algorithm* baik dari *supervised* maupun *unsupervised learning*. Lisensi tersedia secara gratis baik untuk penggunaan akademis maupun komersial. Scikit-Learn dituliskan dalam bahasa Python dan dibangun dari SciPy (Scientify Python) dan Cython. Modul SVM diimplementasikan menggunakan *wrapper* LibSVM/LibLinear. Scikit-Learn menyediakan berbagai algoritma untuk klasifikasi, regresi dan *clustering*.

Tujuan pengembangan *library* ini adalah tingkat keandalan dan dukungan yang diperlukan dalam produksi sistem. Ini berarti pengembangan berfokus terhadap kemudahan penggunaan, kualitas kode, dokumentasi dan performa. Saat ini, Scikit-Learn merupakan *library* yang paling banyak digunakan oleh praktisi dan peneliti yang menggunakan bahasa Python. Pada tugas akhir ini modul yang dimanfaatkan adalah perhitungan TF-IDF, metode klasifikasi (Naive Bayes, KNN, SVM) dan *cross-validation*.

2.8. NLTK

NLTK (Natural Language Toolkit) adalah sebuah *open source library* untuk bahasa pemrograman Python yang mencakup pemrosesan data berupa *human natural language*/bahasa alami manusia [9]. NLTK tersedia secara gratis untuk Windows, Mac OS dan Linux. NLTK digunakan oleh berbagai ahli bahasa, mahasiswa, peneliti dan industri di seluruh dunia. NLTK menyediakan antarmuka untuk 50 korpora dan sumber leksikal beserta *library* untuk pemrosesan klasifikasi, tokenisasi, *stemming*, *tagging*, *parsing* dan *semantic reasoning*. Pada tugas akhir ini modul yang digunakan adalah tokenisasi dan *stemming* (Porter Stemming).

2.9. Penelitian Terkait

Penelitian [3] melakukan percobaan prediksi kepribadian secara otomatis menggunakan fitur linguistik dari teks tertulis dan percakapan. Fitur linguistik yang digunakan adalah kata berbahasa Inggris berdasarkan aplikasi LIWC. Model kepribadian yang digunakan adalah Big Five.

Penelitian [10] melakukan percobaan pencarian kepribadian dari fitur yang ditemukan dari Facebook. Fitur yang dicari adalah linguistik dengan kata berbahasa Inggris berdasarkan aplikasi LIWC, struktural jaringan, aktivitas yang dilakukan dan informasi personal lainnya. Analisis menggunakan aplikasi Weka dengan 2 algoritma, M5 Rules dan Gaussian Processes.

Penelitian [11] menggunakan metode Naive Bayes untuk menentukan kepribadian dari sebuah teks yang dituliskan oleh seseorang. Tulisan yang dibuat adalah deskripsi diri sendiri yang akan digunakan untuk mencari kepribadian dan kemudian dicari pasangan pada situs pencarian jodoh. Bahasa yang digunakan adalah bahasa Indonesia. Model kepribadian yang digunakan adalah Four Temperaments yaitu *Sanguine, Choleric, Melancholic, Phlegmatic.*

Penelitian [12] merupakan kumpulan kontribusi dari beberapa tim dan peneliti yang melakukan penelitian pada prediksi kepribadian melalui sosial media. *Dataset* yang digunakan pada tugas akhir mengacu pada penelitian ini.

BAB III
ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun. Perancangan meliputi perancangan data, perancangan proses dan perancangan antarmuka.

3.1. Arsitektur Umum Sistem

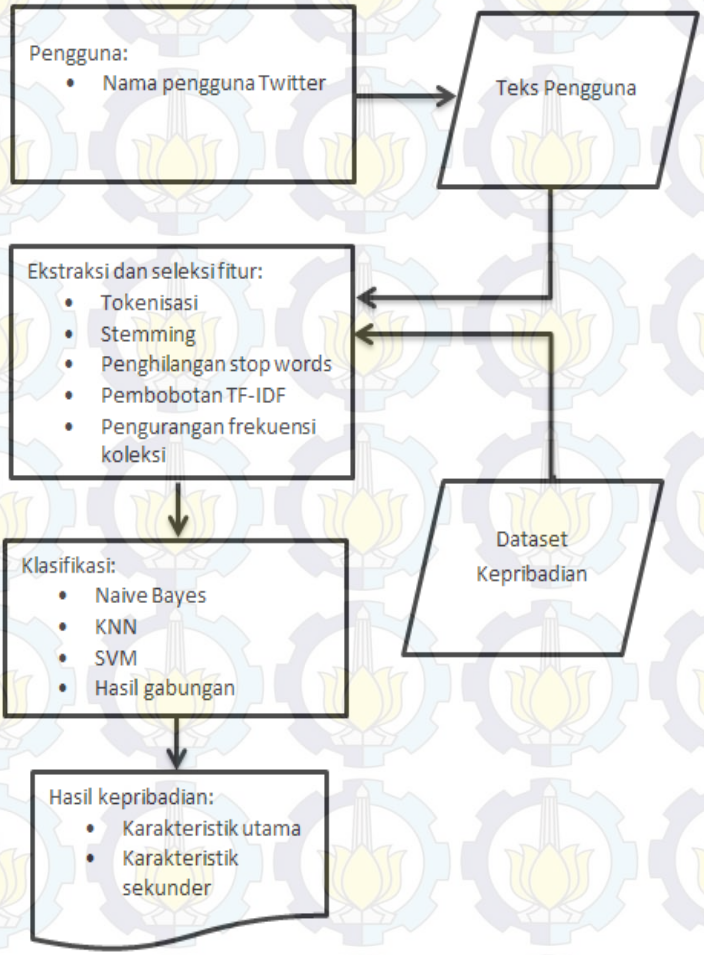
Permasalahan utama yang diangkat dalam pembuatan tugas akhir ini adalah bagaimana mengklasifikasikan tulisan-tulisan yang dibuat oleh pengguna media sosial terhadap tipe kepribadian yang ada. Proses yang terlibat antara lain proses pengambilan teks dari pengguna media sosial Twitter dan klasifikasi untuk memprediksikan tulisan yang didapat berdasarkan data pelatihan yang digunakan. Diagram alir sistem ditunjukkan pada Gambar 3.1.

Sistem akan mengambil data teks dari pengguna Twitter. Pengambilan berdasarkan nama pengguna/*username* yang dimasukkan. Data teks berupa gabungan dari beberapa *tweet* terakhir dari pengguna tersebut yang dijadikan satu *string* panjang. Data teks kemudian diubah menjadi data vektor. Proses klasifikasi dilakukan dengan menghitung dan membandingkan dengan *dataset* yang sudah terdapat di sistem. Sistem kemudian memberikan hasil prediksi kepribadian.

Sistem yang akan dibuat berupa aplikasi berbasis web. Fungsi yang akan dikembangkan adalah pengklasifikasian tulisan berdasarkan jenis kepribadian. Keluaran dari aplikasi berupa prediksi kepribadian pengguna.

3.2. Perancangan Data

Penjelasan tahap perancangan data dibagi menjadi beberapa bagian yaitu data masukan, data yang diproses dan data keluaran.

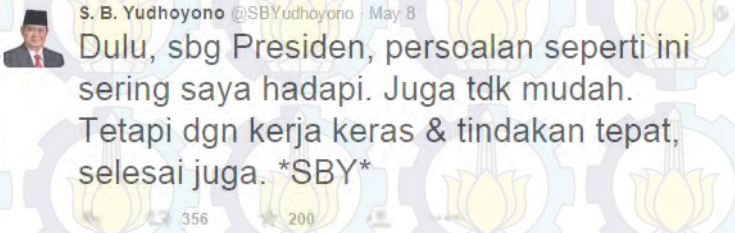


Gambar 3.1 Diagram alir sistem secara umum

3.2.1. Data Masukan

Data masukan merupakan data awal yang akan diproses oleh sistem untuk klasifikasi. Data masukan berupa teks yang diambil dari seorang pengguna Twitter. Contoh tulisan pengguna berupa satu *tweet* dapat dilihat pada Gambar 3.2. Sementara data

masukan untuk suatu pengguna merupakan kumpulan beberapa *tweet* terakhir dari pengguna tersebut yang dijadikan suatu *string* panjang.



Gambar 3.2 Contoh suatu *tweet* atau tulisan pada Twitter

3.2.2. Data Proses

Data proses adalah data yang digunakan selama proses berjalannya sistem. Tabel 3.1 berisi data proses yang digunakan.

Tabel 3.1 Data proses pada sistem

Nama data	Keterangan
Train_data	Kumpulan teks yang digunakan sebagai data pelatihan
Test_data	Teks dari kumpulan tweet yang digunakan sebagai data percobaan
Word_dictionary	Kumpulan kata yang dijadikan fitur untuk perhitungan klasifikasi
Eng_stopwords	Kumpulan kata stop words bahasa Inggris
Ina_stopwords	Kumpulan kata stop words bahasa Indonesia
Data_OVR	*Dataset* yang ditransformasi menjadi *OnevsRest*
MNB_classifiers	Kumpulan classifier untuk metode Naive Bayes
MNB_predict	Hasil prediksi label metode Naive Bayes

KNN_classifiers	Kumpulan *classifier* untuk metode KNN
KNN_score	Hasil skor prediksi metode KNN
KNN_predict	Hasil prediksi label metode KNN
SVM_classifier	Kumpulan *classifier* untuk metode SVM
SVM_score	Hasil skor prediksi metode SVM
SVM_predict	Hasil prediksi label metode SVM
Final_predict	Hasil prediksi yang berupa gabungan ketiga metode
User_primary_characteristics	Karakteristik utama kepribadian dari pengguna
User_secondary_characteristics	Karakteristik tambahan kepribadian dari pengguna
User_freq_words	Kumpulan kata-kata yang sering digunakan pengguna

3.2.3. Data Keluaran

Data keluaran adalah data yang dihasilkan oleh sistem. Data keluaran dari sistem berupa nilai Ya atau Tidak untuk setiap label kepribadian dan karakteristik dari pengguna. Tabel 3.2 adalah contoh dari keluaran sistem.

Tabel 3.2 Contoh data keluaran sistem

Nama data	Nilai
Username	bayuyp
Agreeableness	Ya
Conscientiousness	Tidak
Extraversion	Tidak
Neuroticism	Ya
Openness	Ya
Characteristics_primary	Menghindari hubungan dan sosialisasi terhadap orang lain, dst

Characteristics_secondary	Idealis, Diplomatis, dst
Freq_words	Sekolah, malas, besok, dst

3.3. Perancangan Proses

Pada subbab ini akan dibahas mengenai perancangan proses yang dilakukan ntuk memberikan gambaran secara rinci pada setiap alur implementasi sistem.

3.3.1. Tahap Koleksi Data

Data pelatihan berasal dari myPersonality Project. myPersonality Project adalah aplikasi Facebook yang digunakan untuk tes kepribadian berdasarkan pengisian kuesioner secara *online*. Gambar 3.3 adalah gambar dari isi *dataset* yang didapat. *Dataset* berupa 10000 status dari 250 orang pengguna.
Berikut adalah isi dari *dataset*:

- UserID, merupakan nomor unik pengguna Facebook. Data ini disamarkan untuk privasi pengguna
- Data teks merupakan kumpulan status dari pengguna tersebut.
- Data label didapat dari hasil pengisian kuesioner 100 item IPIP Personality Questionaire oleh pengguna tersebut.
- Label dimodelkan berdasarkan Big Five Personality yaitu AGR (*Agreeableness*), CON (*Conscientiousness*), EXT (*Extraversion*), NEU (*Neuroticism*) dan OPN (*Openness*). Y/Yes berarti pengguna tersebut mempunyai ciri kepribadian dari suatu label.
- Beberapa data *network properties* pengguna seperti *centraliyt* dan *betweenness* dapat diabaikan karena tidak masuk cakupan tugas akhir ini.

USERID	TEXT	AGR	CON	EXT	NEU	OPN
4d035bd3fd8d9595d15cea9e388964be	is celebrating with her...	n	n	y	y	y
4d035bd3fd8d9595d15cea9e388964be	Celebrating 1 year...	n	n	y	y	y
4d035bd3fd8d9595d15cea9e388964be	is smiling	n	n	y	y	y
4d035bd3fd8d9595d15cea9e388964be	There can be no complaints from the Arsenal fans about	n	n	y	y	y
172400f46880b309ca5e97d322bb8f01	I have no excuses, least of all for God. Like all tyrants, he	n	n	n	y	y

Gambar 3.3 *Dataset* orisinal

Data teks yang didapat diproses terlebih dahulu. Semua post dari satu userID digabungkan/*append* menjadi satu baris *string* panjang yang dianggap sebagai satu dokumen. Dilakukan penghilangan 10 pengguna karena hanya terdapat 1-5 kata pada teks yang ditulisnya. *Dataset* akhir berupa 240 dokumen teks untuk tiap orang yang sudah dilabelkan ditunjukkan pada Gambar 3.4.

USER	TEXT	AGR	CON	EXT	NEU	OPN
user1	"...when will the GSS end? is BOSE headphone part of the GSS? w	y	n	n	n	n
user2	"Holy Haruhi, season 2's out?!" A recent online exchange: Other p	n	n	n	y	n
user3	"In my days, we played soccer in the minefield." wants rime with	n	n	y	n	n
user4	"Those who criticize our generation forget who raised it." "In awe	y	y	n	y	y
user5	"we never forget the truth, we only get better at lying to ourselv	n	y	y	n	y
user6	(purposefully contradicting *PROPNAME*) AGHH!! The heat! *op	y	y	n	n	y
user7	YAWN... You no black eye. Boo no fencing tomorrow night. You	n	n	n	n	n

Gambar 3.4 *Dataset* yang sudah diubah

Pengklasikasian bahasa Indonesia menggunakan *dataset* yang sama dengan menerjemahkan seluruh isi menjadi bahasa Indonesia karena tidak tersedianya data kepribadian berdasarkan tulisan asli bahasa Indonesia. Contoh *dataset* yang telah diterjemahkan pada Gambar 3.5. Asumsi yang digunakan adalah pemakaian kata tetap sama walaupun telah dialih-artikan ke bahasa lain. Selama ini penerjemahan otomatis melalui mesin akurat untuk level kata, namun akan menjadi kacau jika menerjemahkan kalimat lengkap. Pendekatan yang dilakukan pada tugas akhir ini adalah per kata sehingga sebagian kata dapat diterjemahkan ke bahasa Indonesia. Solusi ini memiliki keterbatasan yaitu adanya kemungkinan kesalahan dalam pengartian kata yang disebabkan oleh berbagai hal berikut:

- Kehilangan makna karena perbedaan konteks kalimat.
- Terdapat kata bermakna ganda.
- Kata tidak terdapat padanannya dalam bahasa Indonesia.

Data yang akan diprediksi berasal hasil pengambilan teks *tweet* dari seorang pengguna Twitter. Data teks yang diambil berupa *tweet* yang dituliskan langsung oleh sang pengguna tersebut maupun data teks berupa *retweet* (tulisan pengguna lain

yang disebarkan ulang). Jumlah teks yang diambil adalah maksimal 1000 *tweet* terakhir. Kumpulan *tweet* dari pengguna juga dijadikan menjadi sebuah dokumen/*string* panjang. Contoh kumpulan *tweet* yang dijadikan satu dokumen di Gambar 3.6.

USER	TEXT	AGR	CON	EXT	NEU	OPN
user1	akan Headphone wan membayar membenci kehidupan pek	y	n	n	n	n
user2	musim suci pertukaran pria membayangkan sepenuhnya me	n	n	n	y	n
user3	hari bermain sepakbola rime belajar minuman keras high he	n	n	y	n	n
user4	mengkritik generasi lupa mengangkat kagum menyaksikan v	y	y	n	y	y
user5	melupakan kebenaran yang lebih baik berbohong akan tidur	n	y	y	n	y
user6	sengaja bertentangan panas membuka lengan menyambut t	y	y	n	n	y
user7	menguap yay mata hitam boo pagar besok malam menguap	n	y	n	n	y
user8	gambar gelembung sepeda catatan yang diminta korban jatu	n	y	n	n	y

Gambar 3.5 *Dataset* yang diterjemahkan ke bahasa Indonesia

Happy #OpeningDay. This would be huge for working families: #LeadOnLeave Today is the 137th White House Easter Egg Roll. Check out the celebration here: #GimmeFive WATCH: In the weekly address, President Obama talks about the historic understanding reached with Iran. "From my family to yours, Chag Sameach." —President Obama The President is launching an initiative to train 75,000 Americans—including veterans—to join the solar workforce: "We've got to lead by example, invest in the future, train our workers to get jobs in the clean energy economy." —President Obama "We've got to be relentless in our work to grow the economy and create new jobs." —President Obama "Since I took office, solar electricity has gone up twenty-fold." —President Obama #ActOnClimate LIVE: President Obama is in Utah talking about training American workers for clean energy jobs. #ActOnClimate Tune in at 1 p.m. ET to watch President Obama at Hill Air Force Base in Utah discuss the importance of clean energy: Our economy added 126,000 jobs in March, the 61st consecutive month of private-sector job growth. LIVE: President Obama is speaking in Louisville, Kentucky, about the economy. Tune in at 5:50 p.m. ET to watch the President deliver remarks on our economy in Louisville, Kentucky: We need a budget that works for every American—not just the wealthy few. Happening now: The President is delivering a statement on Iran. This 9-year-old girl stood up for what she believes in, and then got a letter back from the President: Read how #Obamacare is a major reason why we've seen an estimated 50,000 fewer preventable patient deaths: Renewable energy investments are up 17 percent globally from 2013. Read more: The budget resolutions Congress passed would devastate programs millions of middle-class families rely upon. Climate change is a global problem. Here's another step towards solving it: #ActOnClimate Get ready for the final sprint. You have until midnight to enter ... Go! This shouldn't be a debate. #ActOnClimate You could talk about anything—even compare jump shots. Enter now: Be one of the people building this grassroots movement from the ground up: Stay calm, but act fast: There's one day left to enter for the chance to meet President Obama. Another nation commits to #ActOnClimate—Mexico announced it would cut carbon pollution 25 percent by 2030: Free flight, free hotel, and a chance to meet the President? Enter today: LIVE: President Obama is speaking about the life and legacy of Senator Ted Kennedy in Boston. "Protecting working Americans' paychecks

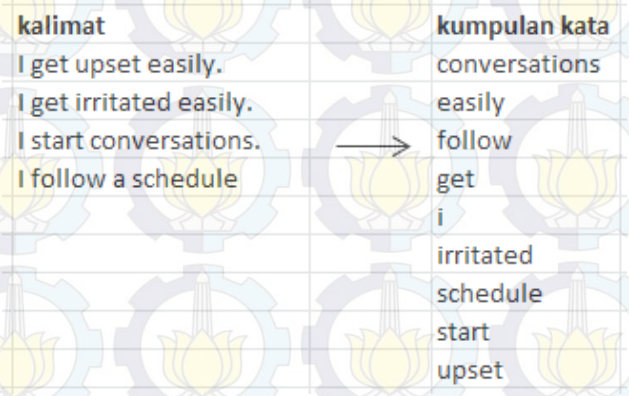
Gambar 3.6 Contoh kumpulan teks dari seorang pengguna

3.3.2. Tahap Ektraksi dan Seleksi Fitur

Pada klasifikasi teks, data yang berupa teks harus terlebih dahulu diubah menjadi data vektor. Data vektor dapat dilihat sebagai vektor dari sebuah kata dimana setiap kata diberikan suatu bobot untuk membantu perhitungan klasifikasi [5, p9]. Model *Bag of Words* adalah model untuk merepresentasikan teks menjadi pecahan kata-kata yang menyusunnya. Penggunaan model ini mengabaikan tata bahasa dan urutan kata, namun tetap menjaga jumlah kemunculan kata. Setelah data diekstraksi atau dijadikan data vektor, proses seleksi fitur dijalankan. Langkah seleksi fitur digunakan untuk mengurangi jumlah fitur/kata yang akan digunakan untuk perhitungan. Beberapa kata yang ada tidak terlalu mempengaruhi proses klasifikasi, bahkan terlalu banyak kata dapat menambah jumlah *noise* sehingga mengurangi akurasi

klasifikasi [5, p15]. Pengurangan fitur diharapkan dapat meningkatkan akurasi dan mengurangi waktu proses.
Berikut langkah ekstraksi fitur dan seleksi fitur yang dilakukan:

- Menghilangkan angka (0….9)
- Menghilangkan tanda baca (, “ ‘: -)
- Menjadikan huruf kapital menjadi huruf kecil (A -> a)
- Tokenisasi: yaitu mengubah kalimat menjadi kumpulan kata tunggal yang menyusunnya. Contoh pada Gambar 3.7.



Gambar 3.7 Contoh tokenisasi

- *Stemming* yaitu mengembalikan sebuah kata menjadi bentuk dasar dengan menghilangkan imbuhan yang ada. *Stemming* diharapkan dapat mengurangi jumlah kata karena kata-kata dengan kata dasar yang sama akan dijadikan menjadi satu fitur [5, p11]. Algoritma *stemming* untuk bahasa Inggris yang digunakan adalah Porter Stemmer. Sementara algoritma untuk bahasa Indonesia adalah algoritma Nazief-Andriani [13]. Contoh proses stemming ditunjukkan pada Gambar 3.8.

kata lengkap		kata dasar
conversations		convers
easily		easili
follow		follow
get	→	get
i		i
irritated		irrit
schedule		schedul
start		start
upset		upset

Gambar 3.8 Contoh proses *stemming*

- Menghilangkan *stop words*. *Stop words* adalah kata-kata yang tidak atau sedikit memiliki arti namun diperlukan dalam struktur gramatikal suatu bahasa [5, p11]. Kata-kata ini hampir selalu ada pada setiap kalimat sehingga tidak berarti untuk proses klasifikasi. Gambar 3.9 merupakan contoh penghilangan *stop words* pada kumpulan kata. Gambar 3.10 merupakan contoh daftar *stop words* yang meliputi:
  - kata ganti (*i, we, you*)
  - bentuk kata kerja (*am, is, are*)
  - kata bantu (*will, can*)
  - preposisi/konjungsi/keterangan (*and, until, from*)
  - kata yang sangat umum (*many, every, just*)
- Pembobotan: untuk setiap kata dilakukan pembobotan untuk perhitungan klasifikasi. Pembobotan menggunakan metode TF-IDF.
- Menghitung nilai *Term Frequency (TF)* sesuai Persamaan 3.1. Contoh perhitungan ditunjukkan Gambar 3.11.

$$tf_t = f_{t,d} \tag{3.1}$$

Keterangan:
tf_t = *term frequency* kata *t*
$f_{t,d}$ = jumlah kemunculan kata *t* pada dokumen *d*

- Menghitung nilai *inverse* dari *Document Frequency (DF)* sesuai Persamaan 3.2. Contoh perhitungan ditunjukkan Gambar 3.12.

$$idf_t = log \frac{N}{df_t} \quad (3.2)$$

Keterangan:
idf_t = *inverse document frequency* kata *t*
N = jumlah total dokumen
df_t = jumlah dokumen dimana terdapat kata *t*

- Menghitung nilai TF-IDF sesuai Persamaan 3.3 Contoh perhitungan ditunjukkan Gambar 3.12.

$$tfidf_t = tf_t \times idf_t \quad (3.3)$$

Keterangan:
$tfidf_t$ = nilai *TF-IDF* kata *t*

Gambar 3.9 Contoh penghilangan *stop words*

contoh stop words	
bahasa Indonesia	bahasa Inggris
ada	a
adalah	about
akan	above
aku	after
anda	again
antara	against
apa	all
apakah	also
atas	am
atau	an
bagaimana	and
bahkan	another

Gambar 3.10 Contoh kata-kata *stop words*

Document	versati	easily	follow	get	i	irritated	chedul	start	upset
I get upset easily.	0	1	0	1	1	0	0	0	1
I get irritated easily.	0	1	0	1	1	1	0	0	0
I start conversations.	1	0	0	0	1	0	0	1	0
I follow a schedule	0	0	1	0	1	0	1	0	0

Gambar 3.11 Contoh perhitungan TF

conve	easily	follow	get	i	irritated	chedul	start	upset
1	2	1	2	4	1	1	1	1

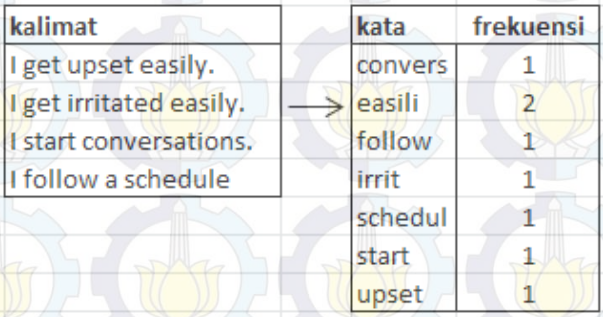
Gambar 3.12 Contoh perhitungan DF

Document	versati	easily	follow	get	i	irritated	chedul	start	upset
I get upset easily.	1.91	1.5	1.91	1.5	1	1.91	1.91	1.91	1.91
I get irritated easily.	0	0.49	0	0.49	0.32	0	0	0	0.63
I start conversations.	0.66	0	0	0	0.34	0	0	0.66	0
I follow a schedule	0	0	0.66	0	0.34	0	0.66	0	0

Gambar 3.13 Contoh pembobotan TF-IDF

- Membatasi jumlah kemunculan. *Collection frequency* adalah jumlah kemunculan sebuah kata pada seluruh dokumen *dataset* yang ada [5, p23]. Jumlah koleksi

diambil dari semua dokumen yang ada tanpa melihat kategori dokumen tersebut. Kita dapat membatasi jumlah fitur dari frekuensi koleksi. Urutkan kata dengan frekuensi koleksi terbanyak dan ambil sebuah angka terbesar. Gambar 3.14 menunjukkan contoh perhitungan frekuensi koleksi suatu kata.



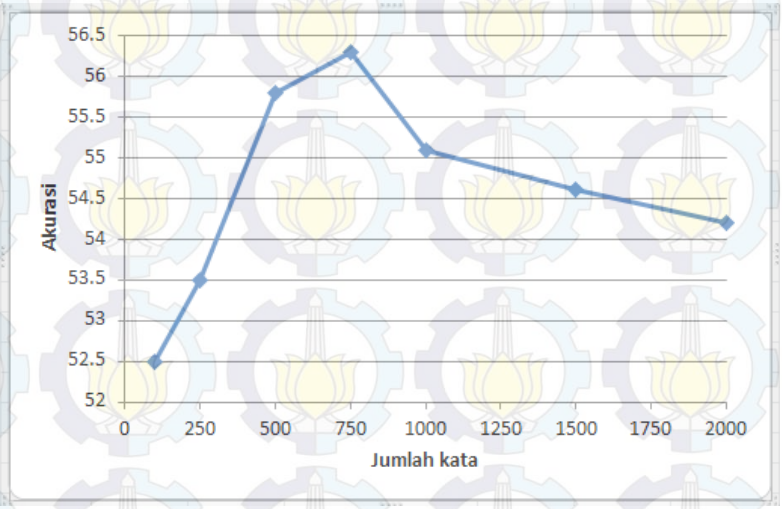
Gambar 3.14 Contoh perhitungan frekuensi koleksi

Pada *dataset* myPersonality memiliki +-10000 kata berbeda. Tabel 3.3 menunjukkan perkiraan minimal kemunculan suatu kata jika diambil *n* kata terbanyak pada *dataset*. Untuk menemukan jumlah kata optimal yang akan digunakan dilakukan percobaan perbandingan akurasi dengan jumlah kata yang diambil. Akurasi diukur menggunakan klasifikasi Multinomial Naive Bayes dengan parameter standar. Hasil percobaan ditunjukkan pada Gambar 3.15. Percobaan dengan 750 kata menjadi parameter maksimal sehingga untuk percobaan selanjutnya digunakan 750 kata yang berarti sebuah kata harus muncul kurang lebih 12 kali pada keseluruhan *dataset*.

Tabel 3.3 Data minimal kemunculan kata

Jumlah kata	Minimal kemunculan
100	+- 70 kali
250	+- 35 kali
500	+- 18 kali

750	+- 12 kali
1000	+- 8 kali
2000	+- 4 kali



Gambar 3.15 Uji coba pencarian jumlah kata

Contoh hasil akhir dari tahap ekstraksi fitur berupa data vektor yang ditunjukkan pada Gambar 3.16. Data vektor dapat dibaca sebagai berikut:

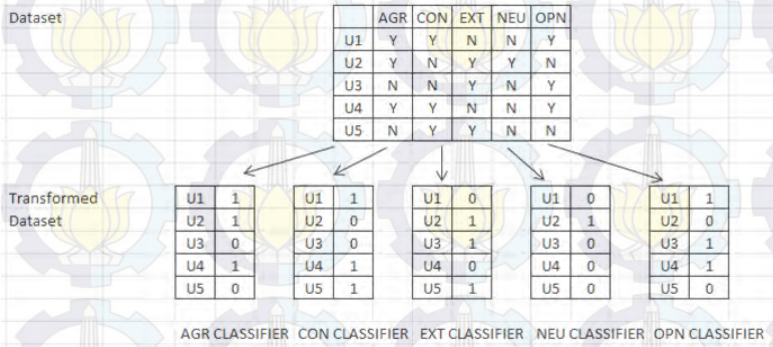
- [x, .., ..] menunjukkan urutan kalimat. 0 berarti kalimat pertama).
- [.., y, ..] menunjukkan urutan kata pada kumpulan kata. Misal: 1 adalah *easily*.
- [.., .., z] menunjukkan bobot kata tersebut dalam kalimat.
- [(0, 1), 0.785] dapat diartikan sebagai berikut: pada kalimat pertama terdapat kata '*easili*' yang mempunyai bobot 0.785.

data teks	kumpulan kata	data vektor
I get upset easily	convers	(0, 3) 0.785
I get irritated easily	easili	(0, 1) 0.619
I start conversations	follow	(1, 1) 1
I follow schedule	upset	(1, 1) 1
		(1, 1)

Gambar 3.16 Hasil ekstraksi fitur berupa data vektor

3.3.3. Tahap Klasifikasi Multi-label

Permasalahan yang diangkat merupakan permasalahan klasifikasi multi-label, yaitu satu orang dapat memiliki satu atau lebih ciri kepribadian atau bahkan tidak memiliki ciri kepribadian sama sekali, maka digunakan metode *multilabel classification*. Metode multi-label yang digunakan adalah *binary relevance* yaitu mentransformasikan label secara biner untuk setiap label terhadap label lainnya dengan asumsi independen atau sering disebut *One vs Rest* [14].



Gambar 3.17 Proses transformasi *multi-label*

Penyelesaian permasalahan adalah dengan membuat *classifier* sebanyak jumlah label yang ada dan dilatih berdasarkan data yang sudah ditransformasikan. Setiap *classifier* merupakan

binary classifier yaitu akan memberi keluaran apakah dokumen percobaan merupakan anggota pada label tersebut atau tidak. Gambar 3.17 adalah contoh transformasi *dataset* pengguna terhadap 5 label kepribadian.

3.3.4. Tahap Klasifikasi Naive Bayes

Metode Naive Bayes bekerja dengan perhitungan probabilitas kelas. Berikut langkah-langkah klasifikasi:

- Hitung probabilitas awal suatu kelas (Persamaan 3.4)

$$P(c) = \frac{N_c}{N} \tag{3.4}$$

Keterangan:
$P(c)$ = probabilitas kelas *c*
N_c = jumlah dokumen pada kelas *c*
N = jumlah total dokumen

- Hitung probabilitas kata (Persamaan 3.5)

$$P(t|c) = \frac{T_{ct} + \propto}{\sum_{t' \in V} (T_{ct'} + \propto)} \tag{3.5}$$

Keterangan:
$P(t|c)$ = probabilitas kata *t* pada kelas *c*
T_{ct} = total bobot kata *t* pada dokumen latih di kelas *c*
V = *vocabulary* atau kumpulan kosa kata
$\propto$ = nilai *smoothing*

- Hitung probabilitas total dan pilih kelas dengan probabilitas terbesar dengan Persamaan 2.2)

Pada perhitungan probabilitas kata (Persamaan 3.5) terdapat parameter $\propto$ atau disebut juga *smoothing* yaitu penambahan suatu nilai pada setiap kata. *Smoothing* dilakukan

untuk menghindari perhitungan nilai nol jika suatu kata tidak terdapat kemunculannya pada suatu dokumen. Nilai $\propto$ merupakan parameter yang dapat diatur dan mempengaruhi probabilitas keluaran dari metode Naive Bayes. Nilai $\propto$ yang digunakan adalah 1 atau yang disebut juga *Laplace smoothing* [8, p360].

3.3.5. Tahap Klasifikasi KNN

Metode KNN bekerja dengan menghitung nilai fungsi jarak (*distance/similarity*) antara data pelatihan dan percobaan. Perhitungan nilai skor menggunakan Persamaan 2.3. Nilai *K* adalah jumlah tetangga yang akan dilakukan *majority vote*. Penentuan nilai *K* sangat krusial. Nilai *K* yang kecil berarti *noise* akan memiliki pengaruh lebih besar pada saat dilakukan *voting*. Menambah nilai *K* menjadi lebih besar akan menambah jumlah *noise* namun pengaruhnya dapat berkurang sehingga ketepatan akan dapat meningkat, meski tidak menjadi jaminan. Nilai *K* yang digunakan adalah 27 yaitu akar dari jumlah fitur [15].

3.3.6. Tahap Klasifikasi SVM

SVM bekerja dengan membuat *hyperplane* atau sekat sebagai pemisah antar kelas.

Berikut langkah klasifikasi:

- Cari *weight vector* optimum dengan Persamaan 3.6 yang memenuhi Persamaan 3.7.

$$\frac{1}{2}w^T w + C\sum_i \xi_i \tag{3.6}$$

$$\{(x_i, y_i)\}, y_i(w^T x_i + b) \geq 1 - \xi_i \tag{3.7}$$

Keterangan:

w = *weight vector*
C = fungsi *loss*
ξ_i = variabel *slack*/kesalahan klasifikasi vektor ke-*i*
x_i = *train vector* ke-*i*

y_i = kelas label *train vector* ke-*i*
b = nilai *bias*

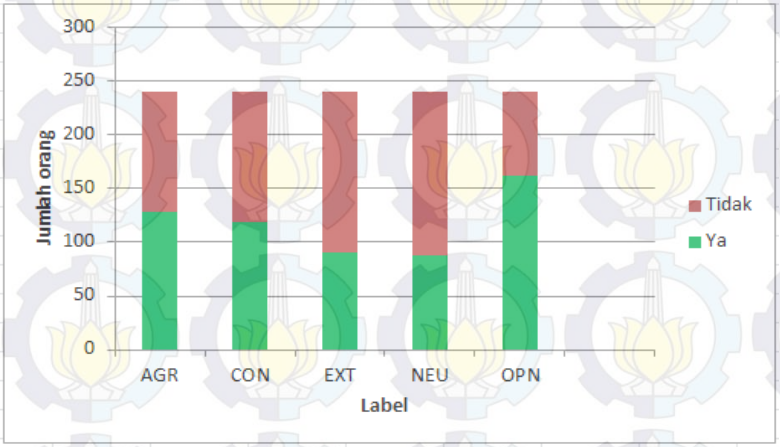
- Tentukan fungsi keputusan atau data letak percobaan pada *hyperplane* menggunakan perkalian *dot product* antara *weight vector* dengan data vektor percobaan seperti dalam Persamaan 2.4.
- Jika fungsi keputusan adalah >= 0 maka dokumen percobaan termasuk label tersebut.

Nilai *C* adalah nilai fungsi *loss* yaitu seberapa besar keinginan menghindari kesalahan klasifikasi. Semakin besar *C* semakin besar juga jarak *hyperplane*. Nilai *C* yang digunakan adalah nilai *default* yaitu 1 [8].

3.3.7. Tahap Tuning Threshold

Distribusi pada *dataset* tidak seimbang. Jumlah orang yang memiliki ciri kepribadian berbeda antar label. Distribusi jumlah orang pada setiap label *dataset* ditunjukkan Gambar 3.20. Sebagai contoh pada label *Openness* terdapat banyak orang yang memilikinya. Sementara label *Neuroticism* memiliki jumlah orang sedikit.

Hal ini akan mempengaruhi perhitungan probabilitas karena jika kelas dengan jumlah orang yang memiliki ciri tersebut jauh lebih banyak dari jumlah yang tidak memiliki, kemungkinan besar bahwa bobot setiap kata akan condong ke arah positif. Untuk mengatasi hal ini dilakukan penyesuaian *threshold* untuk keputusan setiap *classifier*. Secara *default*, *threshold* dari klasifikasi adalah 0.5. Artinya jika pada suatu percobaan probabilitas yang didapatkan lebih dari 0.5 maka akan dikatakan bahwa dokumen percobaan tersebut merupakan bagian dari label. Pengubahan nilai *threshold* ini akan berpengaruh terhadap akurasi prediksi setiap klasifikasi.



Gambar 3.18 Distribusi kepribadian pengguna pemilik kepribadian (240 orang)

Tabel 3.4 Tabel *confusion matrix*

	Prediksi	Prediksi
Aktual	TRUE	FALSE
TRUE	True Positive	False Negative
FALSE	False Positive	True Negative

Secara umum evaluasi klasifikasi menggunakan *confusion matrix* (Tabel 3.4) dengan menghitung Akurasi seperti yang ditunjukkan pada Persamaan 3.8. Namun jika hanya menghitung akurasi maka dapat ditemukan kesalahan yaitu jika *threshold* yang sangat tinggi/sangat rendah. Jika *threshold* terlalu rendah maka sistem akan mengeluarkan lebih banyak prediksi bernilai "Positive" yang merupakan salah satu dari "True Positive" atau "False Positive". Sistem menjadi tidak dapat memprediksi kejadian dimana keluaran seharusnya bernilai "Negative". Untuk menyeimbangkan hal ini ditambahkan 2 parameter yaitu *True Postive Rate* dan *True Negative Rate*.

$$\text{ACC} = \frac{TP+TN}{TP+FP+FN+TN} \quad (3.8)$$

$$\text{TPR} = \frac{TP}{TP+FN} \quad (3.9)$$

$$\text{TNR} = \frac{TN}{FP+TN} \quad (3.10)$$

Keterangan:
ACC = nilai Akurasi
TPR = *True Positive Rate/Sensitivity*
TNR = *True Negative Rate/Specificity*
TP = *True Positive*
TN = *True Negative*
FP = *False Positive*
FN = *False Negative*

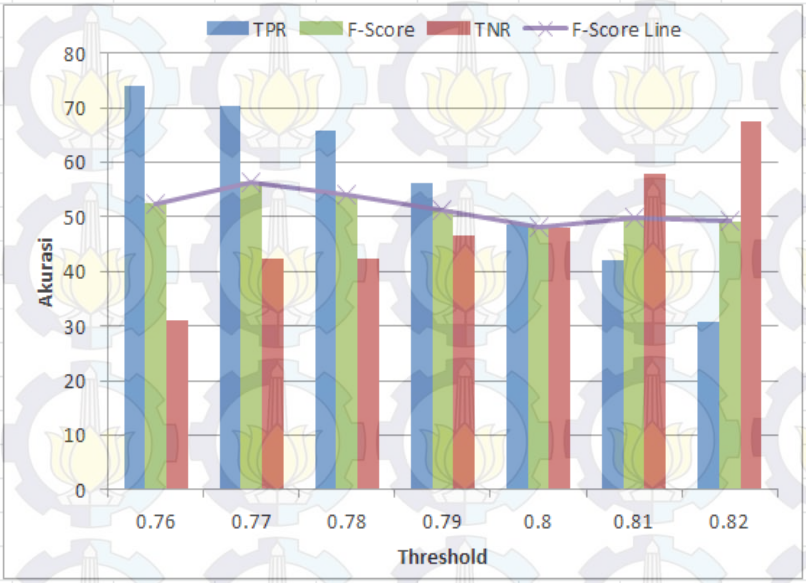
True Positive Rate atau *Sensitivity* (ditunjukkan pada Persamaan 3.9) mengukur bagaimana sistem dapat mendeteksi kebenaran dalam hasil keluaran positif. *True Negative Rate* atau *Specificity* (ditunjukkan pada Persamaan 3.10) mengukur bagaimana sistem dapat mendeteksi kebenaran dalam hasil keluaran negatif. Pada tugas akhir ini nilai *True Negative* diperlukan karena akan digunakan untuk pencarian karakteristik lebih lanjut. Pencarian nilai optimal untuk *threshold* dilakukan dengan parameter metode yang ditentukan pada subbab sebelumnya.

Nilai TPR dan TNR berlawanan dan akan berpotongan pada suatu titik tertentu. Penentuan titik *threshold* diambil dari nilai *F-Score* terbesar dari titik keputusan *classifier*. *F-Score* (ditunjukkan pada Persamaan 3.11) merupakan *mean* dari TPR dan TNR.

$$\text{F-Score} = 2 \times \frac{TPR \times TNR}{TPR + TNR} \quad (3.11)$$

Keterangan:

F-Score = nilai *F-Score* suatu titik



Gambar 3.19 Contoh uji coba pencarian nilai *threshold* label

Gambar 3.19 merupakan contoh grafik pencarian nilai *threshold* optimal. Setiap titik mempunyai nilai TPR, TNR dan F-Score yang berbeda. Pemilihan titik dijatuhkan pada titik yang mempunyai *F-Score* tertinggi sehingga pada contoh ini diambil titik 0.77. Percobaan pencarian *threshold* dilakukan untuk semua label pada setiap metode. Tabel 3.5 dan 3.6 adalah nilai *threshold* optimal yang didapat untuk setiap *classifier* pada *dataset* bahasa Inggris dan bahasa Indonesia.

Tabel 3.5 Tabel nilai *threshold* optimal pada klasifikasi bahasa Inggris

	NB	KNN	SVM
AGR	0.58	0.56	0.25
CON	0.48	0.4	-0.15

EXT	0.34	0.45	-0.1
NEU	0.32	0.33	-0.1
OPN	0.77	0.74	0.25

Tabel 3.6 Tabel nilai *threshold* optimal pada klasifikasi bahasa Indonesia

	NB	KNN	SVM
AGR	0.59	0.59	0.15
CON	0.49	0.44	-0.1
EXT	0.35	0.41	-0.05
NEU	0.3	0.33	-0.05
OPN	0.76	0.74	0.3

Keterangan:
NB = Metode Naive Bayes
KNN = Metode KNN
SVM = Metode SVM
AGR = Label Agreeableness
CON = Label Conscientiousness
EXT = Label Extraversion
NEU = Label Neuroticism
OPN = Label Openness

3.3.8. Tahap Pencarian Hasil Kepribadian

Setiap metode mengeluarkan hasil prediksi yang dapat berbeda antara satu dengan yang lainnya. Untuk menghindari kebingungan dari hasil kesimpulan, dibuat satu hasil keluaran berupa prediksi gabungan yang diambil dari ketiga metode yang ada. Hasil gabungan merupakan hasil *majority vote* dari ketiga metode. Sebagai contoh jika pada suatu label terdapat 2 atau lebih metode yang mengeluarkan prediksi "Ya" maka hasil prediksi akhir label tersebut adalah "Ya". Sebaliknya, jika ada 2 atau lebih prediksi bernilai "Tidak", maka akan hasil metode gabungan adalah "Tidak".

Prediksi FINAL merupakan hasil akhir dari aplikasi dengan menjadikan 3 prediksi masing-masing metode sebagai

pendukung keputusan. Tabel 3.7 menunjukkan kemungkinan keluaran dari prediksi gabungan. Tabel 3.8 merupakan contoh hasil prediksi untuk satu orang. Setiap metode akan menghasilkan 5 keluaran prediksi berdasarkan label Big Five. Akurasi untuk setiap metode dihitung dari jumlah prediksi benar dibagi dengan jumlah label/5 (ditunjukkan pada Persamaan 3.12). Tabel 3.9 merupakan contoh perhitungan akurasi setiap metode berdasarkan contoh pada tabel sebelumnya.

$$Acc\_Method = \frac{N\_True}{N\_Label} \tag{3.12}$$

Keterangan:
Acc_Method = Akurasi tiap metode
N_True = Jumlah prediksi benar
N_Predict = Jumlah prediksi/semua label

Tabel 3.7 Kombinasi hasil prediksi metode gabungan

NB	KNN	SVM	FINAL
Ya	Ya	Ya	Ya
Ya	Ya	Tidak	Ya
Ya	Tidak	Ya	Ya
Ya	Tidak	Tidak	Tidak
Tidak	Ya	Ya	Ya
Tidak	Ya	Tidak	Tidak
Tidak	Tidak	Ya	Tidak
Tidak	Tidak	Tidak	Tidak

Keterangan:
FINAL = Metode gabungan

Tabel 3.8 Contoh hasil prediksi untuk satu orang pengguna

Label	GT	NB	KNN	SVM	FINAL
AGR	Ya	Ya	Ya	Ya	Ya
CON	Tidak	Tidak	Tidak	Ya	Tidak

EXT	Tidak	Tidak	Ya	Ya	Ya
NEU	Tidak	Ya	Tidak	Tidak	Tidak
OPN	Ya	Ya	Ya	Ya	Ya

Keterangan:
GT = Nilai kepribadian berdasar *ground truth*/keputusan ahli

Tabel 3.9 Contoh perhitungan akurasi setiap metode

	NB	KNN	SVM	FINAL
AGR	TRUE	TRUE	TRUE	TRUE
CON	TRUE	TRUE	FALSE	TRUE
EXT	TRUE	FALSE	FALSE	FALSE
NEU	FALSE	TRUE	TRUE	TRUE
OPN	TRUE	TRUE	TRUE	TRUE
N_TRUE	4	4	3	4
N_LABEL	5	5	5	5
Acc_Method	4 / 5 = 80%	4 / 5 = 80%	3 / 5 = 60%	4 / 5 = 80%

Ciri kepribadian utama didapatkan dari pengertian awal label tersebut. Sebagai contoh, pada label *Neuroticism,* nilai "Ya" berarti ciri pengguna tersebut adalah "mudah terpengaruhi oleh emosi negatif". Begitu juga sebaliknya, jika nilai "Tidak" maka pengguna tersebut "tidak mudah terpengaruhi oleh emosi negatif". Tabel keluaran ciri kepribadian utama dapat dilihat pada Lampiran A.1.

Selain ciri kepribadian utama yang didapat dari pengertian label, juga dilakukan pencarian karakteristik sekunder yang didapat dari kombinasi antara 2 label kepribadian utama. Sebagai contoh, seorang pengguna mendapat nilai "Ya" pada label *Neuroticism* dan "Tidak" pada label *Agreeableness*, maka orang tersebut cenderung memiliki karakteristik "Temperamental, Pemarah dan Tidak Sabar". Tabel kombinasi ciri kepribadian sekunder dapat dilihat pada Lampiran A.2.

3.4. Perancangan Antarmuka Perangkat Lunak

Pada subbab ini akan dibahas mengenai perancangan antarmuka perangkat lunak yang bertujuan untuk dapat mempermudah interaksi antara perangkat lunak dengan pengguna.

3.4.1. Halaman Masukan

Halaman masukan (Gambar 3.20) merupakan bagian halaman utama yang ditunjukkan pada pengguna pertama kali. Pengguna dapat memasukkan nama Twitter pada form nama. Tombol prediksi digunakan untuk menjalankan proses klasifikasi. Halaman bahasa Indonesia dan Inggris dapat menggunakan desain yang identik. Spesifikasi atribut antarmuka halaman utama ditunjukkan oleh Tabel 3.10.

Gambar 3.20 Desain halaman utama

Tabel 3.10 Spesifikasi atribut pada halaman utama

No	Nama	Jenis	Keterangan
1	Inggris_Button	Button	Pindah ke halaman bahasa Inggris
2	Indonesia_Button	Button	Pindah ke halaman bahasa Indonesia
3	Input_Name	Input	Ambil nama

			pengguna
4	Predict_Buttion	Button	Jalankan proses klasifikasi

3.4.2. Halaman Hasil

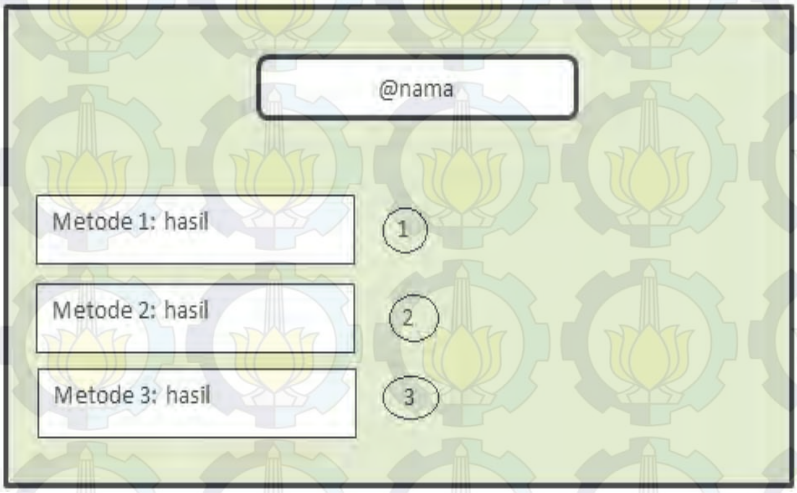
Halaman hasil atau keluaran (desain ditunjukkan pada Gambar 3.21) adalah bagian laman web yang ditampilkan setelah proses klasifikasi dijalankan. Halaman akan berisi prediksi yang merupakan gabungan antara ketiga metode beserta karakteristik kepribadian. Spesifikasi atribut antarmuka halaman hasil ditunjukkan oleh Tabel 3.11.

Gambar 3.21 Desain halaman keluaran

Tombol 'metode' dapat digunakan untuk menampilkan hasil prediksi masing-masing metode (ditunjukkan pada Gambar 3.22). Sedangkan untuk spesifikasi atribut antarmuka halaman detil setiap metode ditunjukkan oleh Tabel 3.12.

Tabel 3.11 Spesifikasi atribut halaman hasil

No	Nama	Jenis	Keterangan
1	User_Name	TextView	Nama user
2	Predict_View	TextView	Hasil prediksi metode gabungan
3	Chara_view	TextView	Hasil karakteristik
4	Method_Button	Button	Tampilkan hasil tiap metode



Gambar 3.22 Desain halaman detil keluaran metode.

Tabel 3.12 Spesifikasi atribut halaman hasil metode

No	Nama	Jenis	Keterangan
1	Method1_View	TextView	Hasil prediksi metode 1
2	Method2_View	TextView	Hasil prediksi metode 2
3	Method3_view	TextView	Hasil prediksi metode 3

BAB IV
IMPLEMENTASI

Bab ini membahas tentang implementasi dari perancangan sistem.

4.1. Lingkungan Implementasi

Dalam pengembangan dan implementasi sistem digunakan perangkat-perangkat pendukung sebagai berikut:

Prosesor : Intel Core i5-3570K @ 3.40GHz
Memori : 8.00 GB
Jenis perangkat : Desktop
Sistem operasi : Microsoft Windows 7 64-bit
Server : XAMPP 1.8.2

4.2. Implementasi Proses

Implementasi proses dilakukan berdasarkan perancangan proses yang sudah dijelaskan pada bab analisis dan perancangan.

4.2.1. Implementasi Proses Pengambilan Teks Twitter

```
1   from twython import Twython
2
3   APP_KEY = 'VFH77cWUfkf51V1Zoq8tH2Ski'
4   APP_SECRET =
'nnrvI9Ar3UKcPlkPr2Pwp0lYK2XaQiqoG95QoWCtS3v0FEoqaw'
5
6   access = Twython(APP_KEY, APP_SECRET,
oauth_version=2)
7   ACCESS_TOKEN = access.obtain_access_token()
8   twitter = Twython(APP_KEY,
access_token=ACCESS_TOKEN)
9
10  username = user
```

```
11  status =
twitter.get_user_timeline(screen_name=username,
count=1000, include_rts=False)
12
13  nl = "\n";
14  long_string = ""
15  list_tweet = []
16  for result in status:
17      sentence = result['text']
18      sentence = "".join(l for l in sentence if l
not in string.punctuation)
19      list_tweet.append(sentence)
20      long_string = long_string + " " + sentence
21
22  test_data = [long_string]
```

Kode Sumber 4.1 Pengambilan data teks Twitter

Pengambilan teks Twitter memanfaatkan Twitter API dengan *library* Python Twython. Data yang diambil memerlukan nama pengguna atau *username*. Jumlah data yang diambil adalah 1000 tweet terakhir termasuk yang berupa *retweet*. Kumpulan tweet yang didapat kemudian dijadikan menjadi satu *string* panjang untuk menjadi data pengujian. Implementasi pengambilan data teks Twitter dapat dilihat pada Kode Sumber 4.1.

4.2.2. Implementasi Proses Import CSV

Data pelatihan myPersonality disimpan dalam .CSV file. Program akan membuka dan membaca isi file CSV. Data teks kemudian dimasukkan ke dalam *train_data*. Sementara label untuk setiap orang ditampung ke dalam variabel sementara, kemudian dimasukkan ke dalam *train_label*. Implementasi pengambilan data pelatihan dapat dilihat pada Kode Sumber 4.2.

```
1    import csv
2
3    train_file =
csv.reader(open('myPersonality.csv', 'r'))
4    train_data = []
5    train_label = []
6    for row in train_file:
7        sentence = row[1]
8       agr = row[2]
9       con = row[3]
10       ext = row[4]
11       neu = row[5]
12       opn = row[6]
13       label = []
14       if agr == 'y':
15           label.append("Agreeableness")
16       if con == 'y':
17           label.append("Conscientiousness")
18       if ext == 'y':
19           label.append("Extraversion")
20       if neu == 'y':
21           label.append("Neuroticism")
22       if opn == 'y':
23           label.append("Openness")
24       train_data.append(sentence)
25       train_label.append(label)
```

Kode Sumber 4.2 Import *dataset* CSV

4.2.3. Implementasi Proses Ekstraksi dan Seleksi Fitur

```
1    from sklearn.feature_extraction.text import
CountVectorizer
2    from sklearn.feature_extraction.text import
TfidfTransformer
3
4    if (lang=='eng'):
5        my_stop_words = set(english_stopwords)
6    elif (lang=='indo'):
```

```
7         my_stop_words = set(indonesia_stopwords)
8
9    vectorizer =
CountVectorizer(analyzer='word',
tokenizer=tokenize, max_features=750,
stop_words=set(my_stop_words))
10   transformer = TfidfTransformer(norm="l2",
smooth_idf=True)
11
12   train_count =
vectorizer.fit_transform(train_data)
13   train_tfidf =
transformer.fit_transform(train_count)
14
15   test_count = vectorizer.transform(test_data)
16   test_tfidf =
transformer.fit_transform(test_count)
```

Kode Sumber 4.3 Ekstraksi dan seleksi fitur

Data teks pelatihan dan pengujian yang telah didapat kemudian diproses terlebih dahulu (Kode Sumber 4.3). Proses meliputi tokenisasi, pembatasan jumlah kata yaitu 750 kata, penghilangan *stop words*, *stemming* dan perhitungan bobot *TF-IDF*.

Proses tokenisasi memecah kalimat menjadi kumpulan kata. Proses penghilangan *stop words* menggunakan daftar kata pada bahasa yang dipilih. Proses *stemming* mengembalikan kata menjadi kata dasar dan dilakukan pada bahasa yang dipilih. Kode Sumber 4.4 menunjukkan implementasi pemanggilan fungsi proses tokenisasi dan Kode Sumber 4.5 menunjukkan cuplikan implementasi *stemming* bahasa Indonesia.

```
1    import nltk
2    import string
3    stemmer = PorterStemmer()
4
5    def stem_tokens(tokens, stemmer):
6        stemmed = []
```

```
7         for item in tokens:
8             stemmed.append(stemmer.stem(item))
9         return stemmed
10
11    def stem_indo_tokens(tokens)
12        stemmed = []
13        for item in tokens:
14            stemmed.append(stemIndo(item))
15        return stemmed
16
17    def tokenize(text):
18        text = ' '.join(s for s in text.split()
if not any(c.isdigit() for c in s))
19        text = "".join(l for l in text if l not
in string.punctuation)
20        tokens = nltk.word_tokenize(text)
21        if (lang=='eng'):
22            stems = stem_tokens(tokens, stemmer)
23        elif (lang=='indo'):
24            stems = stems_indo_tokens(tokens)
25        return stems
26
27    with open("english_stopwords.txt") as
english_stop_file:
28        english_stopwords =
set(word.strip().lower() for word in
english_stop_file)
29
30    with open("indonesia_stopwords.txt") as
indo_stop_file:
31        indonesia_stopwords =
set(word.strip().lower() for word in
indo_stop_file)
```

Kode Sumber 4.4 Pemanggilan tokenisasi dan *stemming*

```
1    def cekKamus(kata):
2        if kata.lower() in kamus_indo:
3            return True
```

```
4
5    def delInflectionSuffixes(kata):
6        sufiks = kata[-3:]
7        if sufiks=='lah' or sufiks=='kah' or
sufiks=='nya' or sufiks=='pun' or sufiks=='tah':
8            kata = kata[:-3]
9        else:
10           sufiks=kata[-2:]
11           if sufiks=='mu' or sufiks=='ku':
12               kata = kata[:-2]
13       return kata
14
15   def delDerivationSuffixes(kata):
16       if kata[-3:] == 'kan':
17           kata = kata[:-3]
18       elif kata[-2:] == 'an':
19           kata = kata[:-2]
20       elif kata[-1:] == 'i':
21           kata = kata[:-1]
22       return kata
23
24   def delDerivationPrefixes(kata):
25       if (kata[:2]=='be' and kata[-1:]=='i'):
26           return kata
27       if (kata[:2]=='di' and kata[-2:]=='an'):
28           return kata
29
30       ...
```

Kode Sumber 4.5 Stemming bahasa Indonesia

4.2.4. Implementasi Proses Transformasi Dataset

```
1    from sklearn.preprocessing import
MultiLabelBinarizer
2
3    target_names = ["Agreeableness",
"Conscientiousness", "Extraversion",
"Neuroticism", "Openness"]
```

```
4    lb = MultiLabelBinarizer()
5    train_label_binary =
lb.fit_transform(train_label)
         proba_all.append(proba)
     num = num + 1
```

Kode Sumber 4.6 Transformasi *dataset*

Sebelum memasuki proses klasifikasi, *dataset* harus ditransformasikan menjadi *binary relevance* seperti pembahasan di subbab 3.3.1. Implementasi transformasi *dataset* dapat dilihat pada Kode Sumber 4.6.

4.2.5. Implementasi Proses Metode Naive Bayes

Implementasi pemanggilan klasifikasi Naive Bayes dapat dilihat pada Kode Sumber 4.7. Proses klasifikasi dimulai dengan membuat *classifier* untuk masing-masing label dengan data pelatihan *dataset* yang sudah diubah menjadi bentuk biner. Setelah dilatih, *classifier* akan memprediksikan keluaran dari suatu dokumen percobaan. Hasil keluaran akan disimpan pada *NB_skor*. Untuk masing-masing keluaran per label dilakukan pengecekan. Jika nilai lebih besar dari *threshold* yang didapat pada subbab 3.3.5 maka keluaran label tersebut diubah menjadi "Ya" (sebelumnya ditentukan sebagai "Tidak" secara *default*). Detil implementasi klasifikasi Naive Bayes pada Kode Sumber 4.8.

```
1   from sklearn.naive_bayes import MultinomialNB
2
3    NB_classifiers = []
4    for i in range (0,5):
5        ovr = train_label_binary[:,i]
6        classifier = MultinomialNB(alpha=1.0)
7        classifier.fit(train_tfidf, ovr)
8        NB_classifiers.append(classifier)
9
10   i = 0
11   for classifier in NB_classifiers:
```

```
12                      ovr_proba                                   =
classifier.predict_proba(test_tfidf)
13      for proba in ovr_proba[:,1]:
14          NB_skor.append(proba)
15          if i == 0:
16              if proba >= 0.58:
17                  NB_AGR = 'Ya'
18          if i == 1:
19              if proba >= 0.48:
20                  NB_CON = 'Ya'
21          if i == 2:
22              if proba >= 0.34:
23                  NB_EXT = 'Ya'
24          if i == 3:
25              if proba >= 0.32:
26                  NB_NEU = 'Ya'
27          if i == 4:
28              if proba >= 0.77:
29                  NB_OPN = 'Ya'
30      i = i + 1
```

Kode Sumber 4.7 Pemanggilan fungsi klasifikasi Naive Bayes

```
1   def fit(self, X, y):
2       """Pembuatan classifier"""
3         X, y = check_arrays(X, y,
sparse_format='csr')
4         y = column_or_1d(y, warn=True)
5         _, n_features = X.shape
6
7         labelbin = LabelBinarizer()
8         Y = labelbin.fit_transform(y)
9         self.classes_ = labelbin.classes_
10        if Y.shape[1] == 1:
11            Y = np.concatenate((1 - Y, Y),
axis=1)
12
13        class_prior = self.class_prior
14
```

```
15          n_effective_classes = Y.shape[1]
16          self.class_count_ =
np.zeros(n_effective_classes, dtype=np.float64)
17          self.feature_count_ =
np.zeros((n_effective_classes, n_features),
18
dtype=np.float64)
19          self._count(X, Y)
20          self._update_feature_log_prob()
21
self._update_class_log_prior(class_prior=class_p
rior)
22          return self
23
24    def _update_class_log_prior(self,
class_prior=None):
25        """Hitung probabilitas kelas"""
26          n_classes = len(self.classes_)
27          if class_prior is not None:
28              self.class_log_prior_ =
np.log(class_prior)
29          elif self.fit_prior:
30              self.class_log_prior_ =
(np.log(self.class_count_)
31                                          -
np.log(self.class_count_.sum()))
32          else:
33              self.class_log_prior_ =
np.zeros(n_classes) - np.log(n_classes)
34
35    def _update_feature_log_prob(self):
36          """Hitung probabilitas kata"""
37          smoothed_fc = self.feature_count_ +
self.alpha
38          smoothed_cc = smoothed_fc.sum(axis=1)
39
40          self.feature_log_prob_ =
(np.log(smoothed_fc)
41                                        -
```

```
np.log(smoothed_cc.reshape(-1, 1)))
42
43   def _joint_log_likelihood(self, X):
44         """Hitung probabilitas total"""
45         X = atleast2d_or_csr(X)
46         return (safe_sparse_dot(X,
self.feature_log_prob_.T)
47                 + self.class_log_prior_)
48
49   def predict_log_proba(self, X):
50         """Return probabiltas dari
pengujian"""
51         jll = self._joint_log_likelihood(X)
52         log_prob_x = logsumexp(jll, axis=1) #
normalize
53         return jll -
np.atleast_2d(log_prob_x).T
```

Kode Sumber 4.8 Detil implementasi klasifikasi Naive Bayes

4.2.6. Implementasi Proses Metode KNN

Implementasi pemanggilan klasifikasi KNN dapat dilihat pada Kode Sumber 4.9. Proses klasifikasi dilakukan bersamaan dengan proses *input* dokumen percobaan. *Classifier* kemudian akan memprediksikan keluaran dari suatu dokumen percobaan. Hasil keluaran akan disimpan pada *KNN_skor*. Untuk masing-masing keluaran per label dilakukan pengecekan. Jika nilai lebih besar dari *threshold* yang didapat pada subbab 3.3.5 maka keluaran label tersebut diubah menjadi "Ya" (sebelumnya ditentukan sebagai "Tidak" secara *default*). Detil implementasi klasifikasi KNN pada Kode Sumber 4.10.

```
1    from sklearn.neighbors import
KNeighborsClassifier
2
3    KNN_classifiers = []
4    for i in range (0,5):
5        ovr = train_label_binary[:,i]
```

```
6       classifier =
KNeighborsClassifier(n_neighbors=27,
algorithm="brute", metric="cosine")
7       classifier.fit(train_tfidf, ovr)
8       KNN_classifiers.append(classifier)
9
10  i = 0
11  for classifier in KNN_classifiers:
12      ovr_proba =
classifier.predict_proba(test_tfidf)
13      for proba in ovr_proba[:,1]:
14          KNN_SKOR.append(proba)
15          if i == 0:
16              if proba >= 0.56:
17                  KNN_AGR = 'Ya'
18          if i == 1:
19              if proba >= 0.4:
20                  KNN_CON = 'Ya'
21          if i == 2:
22              if proba >= 0.45:
23                  KNN_EXT = 'Ya'
24          if i == 3:
25              if proba >= 0.33:
26                  KNN_NEU = 'Ya'
27          if i == 4:
28              if proba >= 0.74:
29                  KNN_OPN = 'Ya'
30      i = i + 1
```

Kode Sumber 4.9 Pemanggilan fungsi klasifikasi KNN

```
1   def _fit(self, n_neighbors=n_neighbors,
algorithm=algorithm, metric=metric)
2                   weights='uniform'):
3
self._init_params(n_neighbors=n_neighbors,
4                             algorithm=algorithm,
5                             leaf_size=leaf_size,
metric=metric, p=p,
6
```

```
metric_params=metric_params, **kwargs)
7
8    def predict_proba(self, X):
9        """Return probabilitas dari
pengujian"""
10       X = atleast2d_or_csr(X)
11       neigh_dist, neigh_ind =
self.kneighbors(X)
12
13       classes_ = self.classes_
14       _y = self._y
15       if not self.outputs_2d_:
16           _y = self._y.reshape((-1, 1))
17           classes_ = [self.classes_]
18
19       n_samples = X.shape[0]
20
21       weights = _get_weights(neigh_dist,
self.weights)
22       if weights is None:
23           weights = np.ones_like(neigh_ind)
24       else:
25           weights[np.isinf(weights)] =
np.finfo('f').max
26
27       all_rows = np.arange(X.shape[0])
28       probabilities = []
29       for k, classes_k in
enumerate(classes_):
30           pred_labels = _y[:, k][neigh_ind]
31           proba_k = np.zeros((n_samples,
classes_k.size))
32
33           for i, idx in
enumerate(pred_labels.T):
34               proba_k[all_rows, idx] +=
weights[:, i] #vote
35
36           normalizer =
```

```
proba_k.sum(axis=1)[:, np.newaxis] #normalize
vote
37              normalizer[normalizer == 0.0] =
1.0
38              proba_k /= normalizer
39
40              probabilities.append(proba_k)
41
42          if not self.outputs_2d_:
43              probabilities = probabilities[0]
44
45          return probabilities
```

Kode Sumber 4.10 Detil implementasi klasifikasi KNN

4.2.7. Implementasi Proses Metode SVM

```
1    from sklearn.svm import LinearSVC
2
3    SVM_classifiers = []
4    for i in range (0,5):
5        ovr = train_label_binary[:,i]
6        classifier = LinearSVC(C = 0.5)
7        classifier.fit(train_tfidf, ovr)
8        SVM_classifiers.append(classifier)
9
10   i = 0
11   for classifier in SVM_classifiers:
12       ovr_proba =
classifier.decision_function(test_tfidf)
13       for proba in ovr_proba:
14           SVM_SKOR.append(proba)
15           if i == 0:
16               if proba >= 0.25:
17                   SVM_AGR = 'Ya'
18           if i == 1:
19               if proba >= -0.15:
20                   SVM_CON = 'Ya'
21           if i == 2:
22               if proba >= -0.1:
```

```
                    SVM_EXT = 'Ya'
            if i == 3:
                if proba >= -0.1:
                    SVM_NEU = 'Ya'
            if i == 4:
                if proba >= 0.25:
                    SVM_OPN = 'Ya'
        i = i + 1
```

Kode Sumber 4.11 Pemanggilan fungsi klasifikasi SVM

Implementasi pemanggilan klasifikasi SVM dapat dilihat pada Kode Sumber 4.11. Implementasai menggunakan *library* LibSVM dan LibLinear. Proses klasifikasi dimulai dengan membuat *classifier* untuk masing-masing label dengan data pelatihan *dataset* yang sudah diubah menjadi bentuk biner. Setelah dilatih, *classifier* akan memprediksikan keluaran dari suatu dokumen percobaan. Hasil keluaran akan disimpan pada *SVM_skor*. Untuk masing-masing keluaran per label dilakukan pengecekan. Jika nilai lebih besar dari *threshold* yang didapat pada subbab 3.3.5 maka keluaran label tersebut diubah menjadi "Ya" (sebelumnya ditentukan sebagai "Tidak" secara *default*). Detil implementasi klasifikasi SVM pada Kode Sumber 4.12.

```
1    def fit(self, X, y):
2        """Pembuatan classifier"""
3          self._enc = LabelEncoder()
4          y_ind = self._enc.fit_transform(y)
5          X = atleast2d_or_csr(X,
dtype=np.float64, order="C")
6
7          self.class_weight_ =
compute_class_weight(self.class_weight,
8
self.classes_, y)
9
```

```
10          y_ind = np.asarray(y_ind,
dtype=np.float64).ravel()
11          self.raw_coef_ =
liblinear.train_wrap(X, y_ind,
12          sp.isspmatrix(X),
13          self._get_solver_type(),
14          self.tol, self._get_bias(),
15          self.C,
16          self.class_weight_,
17          rnd.randint(np.iinfo('i').max))
18
19          if self.fit_intercept:
20              self.coef_ = self.raw_coef_[:,:-1]
21              self.intercept_ =
self.intercept_scaling * self.raw_coef_[:, -1]
22          else:
23              self.coef_ = self.raw_coef_
24              self.intercept_ = 0
25
26          return self
27
28    def decision_function(self, X):
29          """Hitung jarak antara sampel
pengujian ke hyperplane pemisah."""
30          X = self._validate_for_predict(X)
31          X = self._compute_kernel(X)
32          kernel = self.kernel
33
34          dec_func = libsvm.decision_function(
35              X, self.support_,
self.support_vectors_, self.n_support_,
36              self.dual_coef_, self._intercept_,
37              self.probA_, self.probB_,
38
svm_type=LIBSVM_IMPL.index(self._impl),
39              kernel=kernel, degree=self.degree,
cache_size=self.cache_size,
40              coef0=self.coef0,
gamma=self._gamma)
```

```
41
42          return dec_func
```

Kode Sumber 4.12 Detil implementasi klasifikasi KNN

4.2.8. Implementasi Proses Pencarian Hasil Akhir

```
1       agr_count = 0
2       con_count = 0
3       ext_count = 0
4       neu_count = 0
5       opn_count = 0
6
7     if NB_AGR == 'Ya':
8           agr_count = agr_count + 1
9       if KNN_AGR == 'Ya':
10          agr_count = agr_count + 1
11      if SVM_AGR == 'Ya':
12          agr_count = agr_count + 1
13    ...
14
15    if agr_count > 1:
16          FINAL_AGR = 'Ya'
17      if con_count > 1:
18          FINAL_CON = 'Ya'
19      if ext_count > 1:
20          FINAL_EXT = 'Ya'
21      if neu_count > 1:
22          FINAL_NEU = 'Ya'
23      if opn_count > 1:
24          FINAL_OPN = 'Ya'
25
26    if FINAL_AGR=='Ya':
27          trait.append("Senang bekerja sama
dengan orang lain.")
28      if FINAL_AGR=='Tidak':
29          trait.append("Kurang peduli terhadap
orang lain dan kurang memiliki empati.")
30    ...
31
```

```
32  if FINAL_AGR=='Ya' and FINAL_CON=='Ya':
33          char.append("Sopan")
34      if FINAL_AGR=='Ya' and FINAL_CON=='Tidak':
35          char.append("Bersahaja")
36      if FINAL_AGR=='Ya' and FINAL_EXT=='Ya':
37          char.append("Ramah")
38      if FINAL_AGR=='Ya' and FINAL_EXT=='Tidak':
39          char.append("Tidak hati")
40  ...
```

Kode Sumber 4.13 Implementasi pencarian prediksi akhir

Kode Sumber 4.13 merupakan cuplikan implementasi hasil prediksi akhir yang merupakan gabungan antara ketiga metode. Dilakukan *majority vote* pada hasil prediksi masing-masing metode. Prediksi dengan nilai terbanyak (atau lebih dari 1) adalah hasil keputusan akhir. Kemudian karakteristik dicari dari hasil prediksi label dan juga kombinasi antara 2 hasil label kepribadian.

4.3. Implementasi Antarmuka

Pada subbab ini dijelaskan implementasi tampilan antarmuka yang telah dibahas pada subbab 3.4.2. Antarmuka berupa laman web. Implementasi dibuat dalam kode HTML dan CSS. Gambar 4.1 merupakan tampilan halaman utama dimana pengguna akan memasukkan nama.

Gambar 4.2 merupakan tampilan halaman hasil dimana akan ditampilkan hasil prediksi beserta karakteristik pengguna. Tombol ‘Metode’ digunakan untuk membuka tampilan hasil prediksi untuk masing-masing metode (ditunjukkan pada Gambar 4.3).

Gambar 4.1 Halaman utama

Gambar 4.2 Halaman hasil gabungan dan karakteristik

NAIVE BAYES

- Agreeableness: ?
- Conscientiousness: ?
- Extraversion: ?
- Neuroticism: ?
- Openness: ?

K-NEAREST NEIGHBORS

- Agreeableness: ?
- Conscientiousness: ?
- Extraversion: ?
- Neuroticism: ?
- Openness: ?

SUPPORT VECTOR MACHINE: ?

- Agreeableness: ?
- Conscientiousness: ?
- Extraversion: ?
- Neuroticism: ?
- Openness: ?

Gambar 4.3 Halaman hasil metode

BAB V
PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada modul yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap hasil metode dan kebutuhan fungsionalitas sistem.

5.1. Lingkungan Pengujian

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi metode yang sudah dibuat. Lingkungan uji coba yang digunakan meliputi perangkat keras dan perangkat lunak sebagai berikut:

Prosesor	: Intel Core i5-3570K @ 3.40GHz
Memori	: 8.00 GB
Jenis perangkat	: Desktop
Sistem operasi	: Microsoft Windows 7 64-bit
Server	: XAMPP 1.8.2

5.2. Data Uji Coba

Data pelatihan yang digunakan dalam uji coba adalah *dataset* myPersonality yang berisi data teks dari 240 pengguna yang sudah dilabelkan terhadap kepribadian masing-masing orang.

5.3. Skenario Pengujian

Uji coba dilakukan dengan parameter optimal yang didapat dari subbab 3.3.5.

5.3.1. Pengujian Internal Dataset Bahasa Inggris

Tabel 5.1, 5.2, 5.3 merupakan hasil Akurasi, TPR dan TNR dari pengujian 10-*cross validation* terhadap *dataset* myPersonality berbahasa Inggris kepada masing-masing label dari

setiap metode. *Cross-validation* dilakukan sebanyak 10 kali dengan pembagian 90% data pelatihan dan 10% data percobaan.

Rentang akurasi dari semua *classifier* adalah 52.25% hingga 64.92%, dengan performa terbaik pada label *Openness* dari metode KNN. Akurasi ketiga metode secara *macro-average* cukup rendah yaitu 59-60%. Perbedaan yang sedikit membuktikan bahwa tidak ada metode yang unggul secara signifikan. Jika dilihat dari tingkat akurasi maka metode Naive Bayes sedikit lebih baik dari SVM dan KNN dengan akurasi 60.63%. Sementara itu, SVM mendapatkan nilai standar deviasi terkecil (±1.54) dibanding metode lainnya yang menunjukkan bahwa nilai akurasi yang didapat lebih tepat/akurat.

Parameter TPR dan TNR dijadikan sebagai pengukuran sekunder. Nilai rata-rata TPR secara *macro-average* berkisar antara 67-70% dan TNR berkisar antara 48-51% menunjukkan bahwa prediksi sudah cukup seimbang (tidak terlalu *bias* ke positif atau negatif *rate*). Perbedaan nilai TPR dan TNR untuk setiap metode tidak begitu signifikan sehingga penilaian dapat dikembalikan pada tingkat akurasi.

Tabel 5.1 Akurasi percobaan *dataset* bahasa Inggris

	NB	KNN	SVM
AGR	61.33 ± 1.89	57.38 ± 1.67	61.88 ± 1.97
CON	62.87 ± 1.54	63.00 ± 1.68	60.75 ± 1.16
EXT	57.75 ± 1.26	61.46 ± 2.11	59.99 ± 1.75
NEU	57.54 ± 1.9	52.25 ± 1.86	57.3 ± 1.38
OPN	63.67 ±1.54	64.92 ± 2.49	58.2 ± 1.29
AVG	**60.63 ± 1.64**	**59.80 ± 1.98**	**59.62 ± 1.54**

Keterangan:
AVG = Rata-rata

Tabel 5.2 *True Positive Rate dataset* bahasa Inggris

	NB	KNN	SVM
AGR	63.89 ± 3.8	64.65 ± 4.34	67.69 ± 2.34
CON	73.23 ± 4.19	71.08 ± 2.57	75.25 ± 3.12
EXT	69.19 ± 3.14	66.13 ± 4.0	65.67 ± 3.88
NEU	63.19 ± 2.07	73.26 ± 3.78	69.58 ± 2.47
OPN	76.03 ± 2.31	76.57 ± 3.66	61.77 ± 2.34
AVG	**69.11 ± 3.21**	**70.34 ± 3.66**	**67.99 ± 2.89**

Tabel 5.3 *True Negative Rate dataset* bahasa Inggris

	NB	KNN	SVM
AGR	60.94 ± 4.5	49.25 ± 4.3	55.94 ± 2.17
CON	54.08 ± 3.75	54.72 ± 4.52	46.02 ± 2.89
EXT	49.98 ± 3.43	59.95 ± 3.52	56.94 ± 2.2
NEU	55.79 ± 3.98	38.06 ± 3.78	50.10 ± 2.16
OPN	39.15 ± 2.0	38.90 ± 6.03	51.13 ± 15.12
AVG	**51.99 ± 3.63**	**48.17 ± 4.52**	**51.93 ± 3.93**

5.3.2. Pengujian Internal Dataset Bahasa Indonesia

Tabel 5.4, 5.5, 5.6 merupakan hasil Akurasi, TPR dan TNR dari 10-*cross validation* terhadap *dataset* myPersonality yang diterjemahkan ke bahasa Indonesia kepada masing-masing label dari setiap metode. *Cross-validation* dilakukan sebanyak 10 kali dengan pembagian 90% data pelatihan dan 10% data percobaan.

Rentang akurasi dari semua *classifier* adalah antara 51.5% hingga 72.29%, dengan performa terbaik pada label *Openness* dengan metode NB. Akurasi ketiga metode secara *macro-average* cukup rendah yaitu 58-60%. Perbedaan yang sedikit membuktikan bahwa tidak ada metode yang unggul secara signifikan. Jika dilihat dari tingkat akurasi maka metode Naive Bayes sedikit lebih baik dari SVM dan KNN dengan akurasi 60.06%. Sementara itu SVM mendapatkan nilai standar deviasi

terkecil (±1.76) dibanding metode lainnya yang menunjukkan bahwa nilai akurasi yang didapat lebih tepat/akurat.

Parameter TPR dan TNR dijadikan sebagai pengukuran sekunder. Nilai rata-rata TPR secara *macro-average* berkisar antara 69-73% dan TNR berkisar antara 43-48% menunjukkan bahwa prediksi cukup seimbang (tidak terlalu *bias* ke positif atau negatif *rate*). Perbedaan nilai TPR dan TNR untuk setiap metode tidak begitu signifikan sehingga penilaian dapat dikembalikan pada tingkat akurasi.

Tabel 5.4 Akurasi *dataset* bahasa Indonesia

	NB	KNN	SVM
OPN	63.21 ± 1.84	60.71 ± 2.48	62.95 ±1.2
CON	58.67 ± 2.1	61.08 ± 3.19	60.35 ± 1.38
EXT	51.5 ± 2.02	52.87 ± 2.03	53.21 ± 2.17
AGR	54.63 ± 2.01	54.0 ± 1.83	54.2 ± 2.14
NEU	72.29 ± 2.13	62.83 ± 2.07	65.75 ± 1.69
AVG	**60.06 ± 2.02**	**58.3 ± 2.37**	**59.29 ± 1.76**

Tabel 5.5 *True Positive Rate dataset* bahasa Indonesia

	NB	KNN	SVM
OPN	68.38 ± 4.05	67.14 ± 4.14	68.69 ± 2.14
CON	74.81 ± 3.32	77.9 ± 3.02	71.02 ± 2.05
EXT	53.16 ± 4.85	71.7 ± 3.52	62.96 ± 4.17
AGR	66.45 ± 6.42	76.79 ± 4.63	74.27 ± 5.10
NEU	85.15 ± 3.11	72.35 ± 2.52	74.14 ± 1.56
AVG	**69.59 ± 4.51**	**73.18 ± 3.64**	**70.22 ± 3.3**

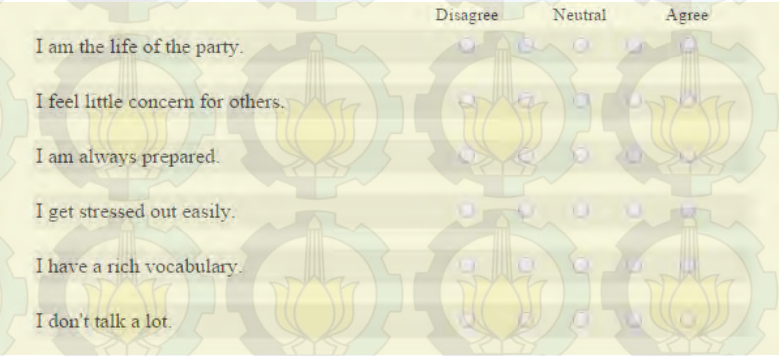
Tabel 5.6 *True Negative Rate dataset* bahasa Indonesia

	NB	KNN	SVM
OPN	59.75 ± 3.71	54.25 ± 5.22	56.6 ± 2.83
CON	42.48 ± 4.77	42.74 ± 4.88	48.7 ± 2.39
EXT	52.86 ± 5.23	41.00 ± 2.36	47.62 ± 2.68
AGR	48.69 ± 2.32	38.86 ± 2.72	40.62 ± 4.07

NEU	40.9 ± 5.74	42.65 ± 2.57	46.97 ± 5.34
AVG	**48.93 ± 4.52**	**43.9 ± 3.55**	**48.11 ± 3.63**

5.3.3. Pengujian Eksternal dengan Responden

Pada skenario ini dilakukan pengambilan data teks dari akun Twitter dari seorang pengguna. Aplikasi akan mengeluarkan hasil klasifikasi dari teks pengguna. Hasil prediksi keempat metode ini kemudian diadu dengan hasil prediksi kepribadian dari pengisian kuesioner IPIP 50-Item Set of IPIP Big Five Factor Markers [16]. Pertanyaan berupa suatu pernyataan yang berhubungan dengan ciri kepribadian Big Five. Jawaban berupa persetujuan dalam skala 1-5 (1 = sangat tidak setuju, 5 = sangat setuju). Contoh daftar pertanyaan pada kuesioner IPIP 50-Item Set dapat dilihat pada Gambar 5.1. Sementara contoh hasil prediksi kepribadian berdasar kuesioner dapat dilihat pada Gambar 5.2.



Gambar 5.1 Contoh daftar pertanyaan pada kuesioner IPIP 50-Item

Responden terdiri dari 40 orang. Responden yang dipilih harus memiliki akun Twitter dengan jumlah *tweet* minimal sebanyak 1000. Responden akan mengisi kuesioner dan melaporkan hasilnya. Sementara sistem akan mengambil teks dari akun Twitter responden untuk diklasifikasikan. Hasil akurasi dari

percobaan responden dapat dilihat pada Tabel 5.7. Detil akurasi pengujian untuk setiap orang dapat dilihat pada Lampiran A.3. Dari percobaan tersebut, metode gabungan mendapat tingkat akurasi terbaik yaitu 65%. Naive Bayes menjadi yang terbaik diantara metode pendukung dengan akurasi 63%, diikuti oleh SVM dan KNN.

Gambar 5.2 Contoh hasil kepribadian dari kuesioner

Tabel 5.7 Akurasi percobaan terhadap 40 responden

	ACC
NB	63%
KNN	60%
SVM	61%
FINAL	65%

Keterangan:
NB = Metode Naive Bayes
KNN = Metode KNN
SVM = Metode SVM

FINAL = Metode gabungan

5.3.4. Pengujian Fungsionalitas Aplikasi

Uji coba fungsionalitas merupakan sebuah pengujian yang dilakukan terhadap jalannya fungsi-fungsi utama pada sistem yang telah dibuat. Halaman depan, ditunjukkan pada Gambar 5.3, adalah halaman utama yang digunakan untuk memasukkan nama pengguna untuk diambil teks Twitter-nya.

Gambar 5.3 Halaman utama dan fungsi memasukkan nama

Gambar 5.4 Hasil prediksi beserta karakteristik

Gambar 5.5 Detil prediksi untuk setiap metode

Gambar 5.4 menunjukkan hasil prediksi akhir dan karakteristik dari pengguna. Tombol "Metode" akan menunjukkan detil prediksi untuk tiap metode dan juga kata-kata yang sering digunakan pengguna tersebut (ditunjukkan pada Gambar 5.5). Dapat disimpulkan bahwa aplikasi berjalan dengan baik dan memenuhi kebutuhan keluaran yang diinginkan.

5.4. Evaluasi Pengujian Hasil Klasifikasi

Dari hasil scenario uji coba yang dilakukan rata-rata akurasi tiap metode yang didapat berkisar antara 58-63%. Dengan selisih sedikit, metode Naive Bayes menjadi yang terbaik diantara ketiga metode yang diuji, diikuti oleh SVM dan KNN.

Metode Naive Bayes merupakan metode perhitungan probabilitas sederhana yang mudah dimengerti. Walaupun sederhana namun tetap merupakan yang terbaik dalam klasifikasi teks. Dalam ketiga percobaan, Naive Bayes selalu mendapatkan akurasi tertinggi diantara ketiga metode yang diuji.

Metode SVM merupakan metode yang lebih rumit dibanding MNB dan KNN. SVM memanfaatkan garis pemisah untuk melakukan perhitungan. Dalam pengujian, hasil yang didapat tidak lebih baik dari Naive Bayes. Pada *dataset* yang digunakan, identifikasi kelas setiap kata kurang begitu tepat sehingga dapat mengurangi akurasi metode SVM.

Metode KNN merupakan metode yang menghitung perbedaan antar data pelatihan dan data pengujian. Metode KNN menjadi metode dengan tingkat akurasi terendah. Dugaan penyebab rendahnya nilai metode KNN adalah karena kesulitan dalam penentuan nilai *K* yang optimal. Jumlah nilai *K* sangat menentukan hasil akhir karena akan dilakukan perhitungan probabilitas pada *K* sampel. Ini berbeda dengan MNB dan SVM yang menggunakan perhitungan murni pada fitur yang ada.

Sementara itu metode gabungan yang merupakan prediksi akhir dari aplikasi mendapatkan hasil yang terbaik pada pengujian responden yaitu 65%. Penggunaan *voting* pada hasil akhir dari ketiga metode pendukung dapat mengurangi kesalahan klasifikasi. Pada kasus dimana salah satu metode kurang tepat dalam melakukan klasifikasi, maka akan dapat tertutupi oleh kedua metode lainnya jika prediksi dari metode lain tersebut adalah benar.

5.5. Evaluasi Hasil Kepribadian

Hasil akhir dari aplikasi berupa prediksi untuk masing-masing faktor pada Big Five Personality. Setiap ciri kepribadian Big Five bersifat independen terhadap ciri lainnya sehingga semua orang sebenarnya memiliki kelima faktor kepribadian tersebut, namun dengan tingkatan yang berbeda. Pada tugas akhir ini, keluaran dari sebuah ciri kepribadian adalah berupa keputusan "Ya", yang berarti pengguna tersebut memiliki ciri kepribadian tersebut secara dominan/tinggi dan "Tidak" yang berarti pengguna kurang/sedikit memiliki ciri kepribadian tersebut.

Ciri kepribadian memengaruhi perilaku seseorang. Kumpulan ciri-ciri ini menentukan individu untuk bertindak

dengan cara tertentu dalam situasi tertentu. Ciri kepribadian biasanya konsisten dan tetap dimiliki oleh orang tersebut seumur hidup namun perilaku tetap dapat berubah melalui proses adaptasi.

Perlu ditekankan bahwa nilai kepribadian yang didapat bukanlah bersifat positif maupun negatif. Setiap ciri kepribadian yang dimiliki seseorang mempunyai kelebihan dan kekurangannya masing-masing. Sebagai contoh, orang dengan ciri *Agreeableness* akan lebih disukai daripada yang tidak memiliki. Akan tetapi orang dengan *Agreeableness* kurang berguna pada situasi yang memerlukan keputusan sulit dan obyektif sehingga orang yang tidak memiliki ciri *Agreeableness* mempunyai kesempatan lebih baik untuk menjadi ilmuwan, kritikus atau tentara.

Dengan memahami setiap faktor dari Big Five akan membantu pengguna memahami ciri kepribadian yang dimilikinya sendiri atau orang lain yang dicari. Berikut adalah ciri kepribadian beserta, sifat-sifat yang dominan beserta kelebihan dan kekurangannya.

- *Extraversion*

  Orang dengan ciri kepribadian ini atau disebut *Extrovert* memiliki sifat mudah berteman, menyukai keramaian, suka berbicara, ingin menjadi pemimpin, energetik, bersemangat, mencari kesenangan, selalu dalam *positive mood* dan senang menarik perhatian orang lain. Sementara orang tanpa kepribadian ini atau disebut *Introvert* memiliki sifat lebih suka menyendiri, menghindari keramaian, tidak senang berbicara, santai dan kurang bersemangat. Sifat-sifat yang ada pada *Introvert* tidak harus diartikan sebagai rasa malu, depresi, tidak bersahabat ataupun arogan. Pada kenyataannya *Introvert* memang memilih untuk tidak mencari hubungan keluar akan tetapi tetap menyenangkan jika didekati terlebih dahulu.
- *Agreeableness*

Orang dengan ciri kepribadian ini memiliki sifat mudah bergaul, mudah memercayai orang lain, membantu orang lain atas inisiatif sendiri, senang bekerja sama, dapat berkompromi, lebih mementingkan kepentingan bersama, rendah hati dan memiliki empati. Orang tanpa ciri kepribadian ini memiliki sifat kurang senang membantu, mementingkan kepentingan sendiri, skeptis terhadap orang lain, kurang kooperatif, bangga pada diri sendiri dan obyektif. Orang dengan ciri *Agreeableness* tentu saja lebih disukai oleh orang lain. Di sisi lain orang dengan *Agreeableness* kurang begitu berguna pada situasi yang memerlukan keputusan obyektif dan sulit sehingga orang tanpa *Agreeableness* mempunyai kesempatan lebih baik untuk menjadi ilmuwan, kritikus atau tentara.

- *Conscientiousness*

  Orang dengan kepribadian ini memiliki sifat percaya diri akan sukses, hidup teratur, bertanggung jawab pada pekerjaan dan kewajiban, motivasi untuk sukses tinggi, disiplin diri tinggi dan berpikir sebelum bertindak. Orang tanpa kepribadian ini memiliki sifat kurang percaya diri untuk sukses, hidupnya kurang teratur, kurang dapat diandalkan, kurang bertanggung jawab, malas bekerja, kurang disiplin, berpikir tanpa memertimbangkan alternatif atau konsekuensinya. Kelebihan dari orang dengan *Conscientiousness* adalah menghindari masalah dan mencapai tingkat keberhasilan tinggi melalui perencanaan matang dan ketekunan. Negatifnya adalah mereka bisa jadi merupakan orang yang perfeksionis, *workaholic*, kaku dan membosankan. Sementara itu, orang tanpa *Conscientiousness* dapat diartikan sebagai kurang dapat diandalkan, kurang ambisius dan sering mengalami kegagalan namun santai, tidak kaku dan mudah bergaul.
- *Neuroticism*

Orang dengan ciri kepribadian ini (nilai yang didapat tinggi) memiliki sifat mudah mengalami berbagai emosi negatif seperti cemas, takut, marah, depresi, gugup dan bingung. Mereka juga sensitif terhadap apa yang dipikirkan orang lain serta mudah panik dalam keadaan tertekan. Orang tanpa ciri kepribadian ini memiliki sifat tenang, percaya diri dan stabil secara emosional. Sifat-sifat pada orang dengan *Neuroticism* dapat mengurangi kemampuan untuk berpikir jernih, membuat keputusan dan mengatasi *stress*.

- *Openness*

  Orang dengan ciri kepribadian ini memiliki sifat imajinasi tinggi, menghargai seni, intelek, senang dengan ide baru dan terbuka dengan perubahan. Orang tanpa ciri kepribadian ini memiliki ciri kurang menghargai seni, lebih memilih berhubungan dengan orang daripada suatu gagasan, berpikir secara konservatif dan bertindak secara konvensional. Orang dengan ciri *Openness* lebih cocok pada lingkungan akademis dan seni. Orang tanpa ciri *Openness* lebih baik pada pekerjaan yang berhubungan dengan layanan publik lainnya seperti polisi.

Aspek kepribadian pengguna dapat digunakan dalam beberapa hal dalam kehidupan nyata seperti pertimbangan dalam proses seleksi/rekrutmen pegawai, prediksi penilaian performa kerja serta diagnosa kesehatan dan psikologis seseorang. Psikolog tidak akan mendeskripsikan seseorang apakah orang tersebut mempunyai sifat *Conscientiousness* atau tidak. Mereka akan menggunakan informasi tersebut untuk membuat prediksi-prediksi tertentu tentang kemungkinan perilaku orang tersebut di masa depan.

Sebagai contoh seorang siswa A memiliki nilai sekolah tinggi. Jika siswa tersebut menjalani tes kepribadian, dapat diprediksikan ia akan memiliki ciri kepribadian *Conscientiousness* yang tinggi. Sementara siswa B sering

membolos dan memiliki nilai buruk. Besar kemungkinan nilai *Conscientiousness* siswa B tidaklah setinggi siswa A. Dengan mengetahui informasi ini, kita dapat membuat prediksi-prediksi berupa tindakan yang akan dilakukan di masa yang akan datang. Misalkan siswa A tidak akan mengalami kesulitan pada tingkat studi lebih lanjut, sementara siswa B mungkin kurang cocok untuk meneruskan ke jenjang perguruan tinggi. Tindakan ini akan membantu orang yang bersangkutan untuk mempersiapkan diri dan juga menghindari potensi-potensi yang dapat menghalangi.

Apakah orang dengan *Conscientiousness* tinggi sudah pasti lancar di perguruan tinggi? Begitu pula sebaliknya, apa *Conscientiousness* rendah berarti tidak dapat mengikuti pembelajaran dengan baik? Tentu saja tidak. Ciri kepribadian hanya mengeluarkan nilai prediksi yang kemudian dapat disimpulkan namun tetap saja tidak dapat secara akurat memprediksikan suatu kejadian.

Contoh lain adalah dalam bidang pekerjaan. Pegawai yang baik adalah pegawai dengan *Conscientiousness* tinggi. Pegawai dengan *Extraversion* tinggi yang berarti terbuka terhadap orang lain sehingga cocok pada bidang *sales*/penjualan. Pegawai yang dapat bekerja secara tim dengan baik kemungkinan besar memiliki *Agreeableness* tinggi. Berikut adalah contoh-contoh penggunaan prediksi Big Five pada kehidupan nyata:

- Kenakalan remaja dapat diprediksikan dari nilai *Agreeableness* (rendah) dan *Conscientiousness* (rendah)
- Gangguan psikologis dapat diprediksikan dari nilai *Neuroticism* (tinggi) dan Conscientiousness (rendah).
- Performa akademis dapat diprediksikan dari nilai *Conscientiousness* (tinggi) dan *Openness* (tinggi)
- Penilaian atau perektrutan pegawai dapat diprediksikan dari nilai *Conscientiousness* (tinggi), meski ciri kepribadian lain juga dapat memprediksi pada hal yang lebih spesifik misal *Extraversion* (tinggi) cocok pada bidang *sales* dan manajemen.

- Kesehatan yang baik dan umur panjang dapat dilihat dari nilai *Conscientiousness* (tinggi). Sementara *Agreeableness* (rendah) dan *Neuroticism* (rendah) menandakan adanya faktor risiko kesehatan yaitu rentan depresi.
- Pemilihan teman sebaiknya melihat pada nilai *Ektraversion* (tinggi) dan *Agreeableness* (tinggi).
- Pemilihan kepemimpinan atau *leadership* dapat melihat nilai *Extraversion* (tinggi).
- Bidang yang membutuhkan kreativitas dan intelektual seperti seni dan akademis/penelitian membutuhkan sifat-sifat *Openness* (tinggi).

Beberapa aspek penilaian kepribadian tidak dimasukkan dalam Big Five seperti motivasi, emosi, sikap, kemampuan, konsep diri dan peran sosial. Meski beberapa hal tersebut secara teori atau empiris masih dapat dihubungkan dengan ciri kepribadian Big Five, namun secara konseptual mereka sudah berbeda. Oleh karena itu, profil kepribadian yang didapat dari percobaan ini bukanlah kepribadian yang komprehensif melainkan hanya gambaran sebagian dari kepribadian mereka.

BAB VI
KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan mengenai uji coba serta beberapa saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut:

1. Data teks dari pengguna Twitter berhasil didapatkan dengan memanfaatkan API yang tersedia.
2. Klasifikasi teks berdasarkan kepribadian berhasil diprediksikan dengan mengadunya dengan *dataset* yang sudah dilabelkan.
3. Hasil yang diperoleh dari 3 metode dapat dibandingkan berdasarkan akurasi, *sensitivity* dan *specificity*.
4. Metode Naive Bayes menghasilkan akurasi yang sedikit lebih baik dari metode lainnya dengan tingkat akurasi 63%.
5. Sistem dapat memprediksikan hasil akhir kepribadian pengguna dari teks Twitter dengan tingkat akurasi 65%.

6.2. Saran

Berikut adalah saran-saran untuk perbaikan dan pengembangan sistem di masa yang akan datang:

1. Penggunaan *dataset* yang lebih akurat untuk meningkatkan tingkat akurasi.
2. Penggunaan *dataset* bahasa Indonesia yang diambil langsung dari teks berbahasa asli Indonesia, bukan hasil penerjemahan.
3. Penambahan media sosial lain.
4. Pendekatan semantik dengan mencari makna kata dalam kalimat.

DAFTAR PUSTAKA

[1] Cantandir, I., Fernandez-Tobiaz, I., Belllogin, A., 2013. "Relating Personality Types with User Preferences in Multiple Entertainment Domains". EMPIRE 1st Workshop on Emotions and Personality in Personalized Services.

[2] Goldberg, L.R., Johnson, J.A., Eber, H.W., Hogan, R., Ashton, M.C., Cloninger, C.R.., 2006. "The International Personality Item Pool and the Future of Public Domain Personality Measures". Journal of Research in Personality, 40(1), 84-96.

[3] Mairesse, F., Walker, M., Mehl, M., Moore, R., 2007. "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text". Journal of Artificial Intelligence Research (JAIR). 30(1), 457–500.

[4] Costa, P.T., McCrae, R.R., 1992. "Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI)". Psychological Assessment Resources.

[5] Garnes, Ø. L. 2009. Feature Selection for Text Categorization. Oslo.

[6] Manning, C.D., Raghavan, P., Schutze, H. 2008. Introduction to Information Retrieval. Cambridge UP.

[7] Twitter Developer. 2015. Twitter API Documentation, <URL: https://dev.twitter.com/overview/documentation>

[8] Scikit-Learn Developers. 2015. Scikit-Learn 0.16.1 Documentation,<URL:http://scikit-learn.org/stable/documentation.html/>

[9] NLTK Developers. 2015. NLTK 3.0 Documentation, <URL: http://nltk.readthedocs.org/en/latest/>

[10] Golbeck, J., Robles C., Turner, K., 2011. "Predicting Personality with Social Media". In CHI'11 Extended Abstracts on Human Factors in Computing Systems, pp. 253-262.

[11] Lestari, N.M.A., Putra, I.K.G.D., Cahyawan, A.A.K.A., 2013. "Personality Types Classification for Indonesian Text in Partners Searching Website Using Naive Bayes Methods". International Journal of Software and Informatics Issues.

[12] Celli, F., Pianesi, F., Stillwell, D., Kosinski, M., 2013. "Workshop on Computational Personality Recognition (Shared Task)". In Proceedings of WCPR13, in conjunction with ICWSM-13.

[13] Nazief, B.A.A., Adriani, M., 1996. "Confix-Stripping: Approach to Stemming Algorithm for Bahasa Indonesia," Internal publication, Faculty of Computer Science, Univ. of Indonesia.

[14] Tsoumakas, G., Katakis, I., 2007. "Multi-label Classification: an Overview". International Journal of Data Warehousing & Mining 3 (3): 1–13.

[15] Duda, R.O., Hart, P.E., Stork, D.G. 2001. Pattern Classification, 2nd ed. Wiley.

[16] Goldberg, Lewis R., 1992. "The Development of Markers for the Big-Five Factor Structure". Psychological Assessment 4.1: 26.

LAMPIRAN

Lampiran A.1 Tabel Dimensi Karakteristik Utama Kepribadian

Label	Hasil	Ciri kepribadian
Agreeableness	Ya	Senang bekerja sama dengan orang lain.
Agreeableness	Tidak	Kurang peduli terhadap orang lain dan kurang memiliki empati.
Conscientiousness	Ya	Hidup teratur dan mempunyai motivasi tinggi.
Conscientiousness	Tidak	Santai dan kurang termotivasi untuk sukses.
Extraversion	Ya	Senang berhubungan dan bersosialisasi dengan orang lain.
Extraversion	Tidak	Menghindari hubungan dan sosialisasi terhadap orang lain.
Neuroticism	Ya	Mudah mengalami perubahan suasana hati dan terpengaruh emosi negatif.
Neuroticism	Tidak	Tidak mudah terpengaruh oleh emosi negatif.
Openness	Ya	Menyukai perubahan dan mempunyai keingintahuan tinggi.
Openness	Tidak	Tidak menyukai perubahan dan keingintahuan kurang.

Lampiran A.2 Tabel Dimensi Karakteristik Sekunder Kepribadian

http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/personality-insights/resources/PI-Facet-Characteristics.pdf

Secondary Trait		Primary Trait: Openness		Emotional Range		Extraversion		Conscientiousness		Agreeableness	
		Low	High	Low	High	Low	High	Low	High	Low	High
Agreeableness	High	Simple, dependent	Idealistic, diplomatic, deep, tactful, genial	Patient, relaxed, undemanding, down-to-earth	Emotional, gullible, affectionate, sensitive, soft	Unaggressive, humble, submissive, timid, compliant	Social, energetic, enthusiastic, communicative, vibrant	Unpretentious, self-effacing	Dependable, responsible, reliable, mannerly, considerate		
Agreeableness	Low	Coarse, tactless, curt, narrow-minded, callous	Shrewd, eccentric, individualistic	Unemotional, insensitive, unaffectionate, passionless	Temperamental, irritable, quarrelsome, impatient, grumpy	Skeptical, wary of others, seclusive, uncommunicative, unsociable	Opinionated, forceful, domineering, boastful, bossy	Rash, uncooperative, unreliable, distrustful, thoughtless	Stern, strict, rigid		
Conscientiousness	High	Conventional, traditional	Analytical, perceptive, informative, articulate, dignified	Rational, objective, steady, logical, decisive	Particular, high-strung	Restrained, serious, discreet, cautious, principled	Active, competitive, persistent, ambitious, purposeful			Strict, rigid, stern	Helpful, cooperative, considerate, respectful, polite
Conscientiousness	Low	Shortsighted, foolhardy, illogical, immature, haphazard	Unconventional, quirky	Informal, low-key	Compulsive, nosy, self-indulgent, forgetful, impulsive	Indirect, unenergetic, sluggish, non-persistent, vague	Boisterous, mischievous, exhibitionistic, gregarious, demonstrative			Inconsiderate, impolite, distrustful, uncooperative, thoughtless	Unpretentious, self-effacing
Extraversion	High	Verbose, unscrupulous, pompous	Wordly, theatrical, eloquent, inquisitive, intense	Unselfconscious, weariless, indefatigable	Excitable, wordy, flirtatious, explosive, extravagant			Unruly, boisterous, reckless, devil-may-care, demonstrative	Ambitious, alert, firm, purposeful, competitive	Bullheaded, abrupt, crude, combative, rough	Effervescent, happy, friendly, merry, jovial
Extraversion	Low	Predictable, unimaginative, somber, apathetic, unadventurous	Introspective, meditative, contemplating, self-examining, inner-directed	Unassuming, unexcitable, placid, tranquil	Guarded, fretful, insecure, pessimistic, secretive			Indecisive, aimless, wishy-washy, noncommittal, un-ambitious	Cautious, confident, punctual, formal, thrifty	Cynical, wary of others, reclusive, detached, impersonal	Soft-hearted, agreeable, obliging, humble, lenient
Emotional Range	High	Easily rattled, easily irked, apprehensive	Passionate, excitable, sensual			Guarded, pessimistic, secretive, cowardly	Explosive, wordy, extravagant, volatile, flirtatious	Scatterbrained, inconsistent, erratic, forgetful, impulsive	Particular, high-strung	Critical, selfish, ill-tempered, antagonistic, grumpy	Sentimental, affectionate, sensitive, soft, passionate
Emotional Range	Low	Imperturbable, insensitive	Creative, intellectual, insightful, versatile, inventive			Tranquil, sedate, placid, impartial, unassuming	Confident, bold, assured, uninhibited, courageous	Informal, low-key	Thorough, steady, consistent, self-disciplined, logical	Insensitive, unaffectionate, passionless, unemotional	Generous, pleasant, tolerant, peaceful, flexible
Openness	High			Heartfelt, versatile, creative, intellectual, insightful	Excitable, passionate, sensual	Inner-directed, introspective, meditative, contemplating, self-examining	Expressive, candid, dramatic, spontaneous, witty	Unconventional, quirky	Sophisticated, perfectionistic, industrious, dignified, refined	Shrewd, eccentric, individualistic	Genial, tactful, diplomatic, deep, idealistic
Openness	Low			Imperturbable, insensitive	Easily rattled, easily irked, apprehensive	Somber, meek, unadventurous, passive, apathetic	Verbose, unscrupulous, pompous	Foolhardy, illogical, immature, haphazard, lax	Traditional, conventional	Coarse, tactless, curt, narrow-minded, callous	Dependent, simple

Lampiran A.3 Detil Akurasi Pengujian pada Responden

	Akurasi			
Nama Twitter	**NB**	**KNN**	**SVM**	**FINAL**
_rzld	80%	40%	100%	80%
a_tev	40%	60%	60%	80%
adhiwiyono	40%	40%	40%	40%
ahmadfauzi__	80%	60%	40%	60%
andririnaldis	80%	60%	80%	80%
besta_	80%	40%	40%	40%
bskr_	40%	80%	100%	80%
danangpeen	60%	40%	40%	60%
farisme	80%	100%	60%	40%
hernantas	80%	80%	100%	100%
luthfanar	40%	40%	80%	60%
mahardhika_lfc	60%	80%	60%	40%
marchettya	40%	60%	40%	60%
mfarisghani	40%	60%	80%	80%
nbanitama1992	60%	20%	60%	80%
radityabrahmana	60%	80%	40%	40%
rifi166	40%	20%	20%	40%
rimbykamesworo	40%	60%	40%	40%
satriahelmy	40%	60%	40%	60%
ujekfauzy	40%	60%	60%	80%
ardiniocta	80%	60%	80%	100%
baluqio	60%	60%	60%	80%
bimosouza	60%	80%	20%	40%
caksas	60%	40%	40%	40%
Dinahidayanti	60%	100%	100%	80%
farid_dhika	80%	60%	80%	80%
galihKA	60%	60%	60%	40%

Hizkia_Eben	40%	40%	100%	100%
lola_akvyola	60%	60%	80%	100%
marinamarsudip	80%	80%	100%	80%
mchafidm	60%	60%	20%	20%
nosninos	80%	60%	60%	80%
ranahilmy	60%	60%	60%	80%
rezaadzani	100%	80%	60%	60%
riskyky1	80%	60%	80%	80%
rizalharahap	60%	40%	20%	20%
Soe_Hadi	80%	80%	60%	60%
yonditod	80%	80%	80%	60%
Zemzemzemy	80%	40%	60%	80%
ziyuzt	80%	60%	40%	60%
Rata-rata	**63%**	**60%**	**61%**	**65%**

Keterangan:
NB = Metode Naive Bayes
KNN = Metode KNN
SVM = Metode SVM
FINAL = Metode Gabungan
* Setiap metode menguji 5 label kepribadian. Keluaran suatu label dikatakan benar jika hasil prediksi tiap label metode sama dengan hasil prediksi label dari pengisian kuesioner *online*. Pembahasan pada subbab 3.3.6.

Klasifikasi Kepribadian Berdasarkan Tulisan dari Twitter Menggunakan Metode Naive Bayes, KNN dan SVM

Bayu Yudha Pratama, Riyanarto Sarno, Ratih Nur Esti Anggraini
Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: riyanarto@if.its.ac.id

***Abstrak*—Kepribadian merupakan komponen dasar dari perilaku manusia. Kepribadian telah terbukti memengaruhi interaksi dan preferensi seorang individu. Hingga saat ini, untuk mengukur kepribadian mereka diharuskan mengambil tes kepribadian. Media sosial adalah tempat dimana seorang individu mengekspresikan dirinya kepada dunia luar. Tulisan yang dibuat oleh pengguna media sosial dapat dianalisis untuk mendapatkan informasi yang diinginkan. Dalam penelitian ini dilakukan klasifikasi kepribadian berdasar teks yang dituliskan oleh pengguna media sosial Twitter. Bahasa yang digunakan adalah bahasa Indonesia dan bahasa Inggris. Metode klasifikasi yang diimplementasikan adalah Naive Bayes, K-Nearest Neighbors dan Support Vector Machine. Hasil uji coba menunjukkan metode Naive Bayes dengan rata-rata akurasi 63% sedikit mengungguli metode lainnya.**

***Kata Kunci*—Identifikasi kepribadian, K-Nearest Neighbors, Klasifikasi teks, Media sosial, Naive Bayes, Support Vector Machine**

I. PENDAHULUAN

PERSONALITY atau kepribadian adalah kombinasi dari sifat dan tingkah laku seseorang dalam menghadapi berbagai situasi. Kepribadian seseorang dapat memengaruhi pemilihan individu tersebut dalam berbagai hal seperti situs web, buku, musik dan film [1]. Selain itu, kepribadian juga memengaruhi interaksi individu tersebut terhadap orang lain dan lingkungan sekitar. Kepribadian seseorang menjadi salah satu tolok ukur penilaian dalam berbagai bidang seperti seleksi dalam perekrutan pegawai, konseling karir, konseling hubungan/*relationship* maupun konseling kesehatan dan keselamatan kerja.

Selama ini untuk mengetahui kepribadian seseorang, maka orang tersebut harus mengikuti berbagai tes kepribadian. Tes kepribadian dapat berupa deskripsi diri, wawancara maupun observasi yang dilakukan ahli psikologi. Tentunya ini kurang praktis dan cukup merepotkan. Belakangan ini juga dikembangkan tes kepribadian dengan metode kuesioner yang dapat dilakukan pengguna di dunia maya [2], namun tetap saja cara ini kurang praktis karena pengguna masih harus mengisi beberapa pertanyaan. Ciri kepribadian seseorang dapat diperoleh secara otomatis dari teks yang dituliskannya [3]. Pemilihan kata-kata yang sering digunakan dapat menggambarkan kepribadian orang tersebut.

Situs media sosial adalah tempat dimana pengguna merepresentasikan dirinya terhadap dunia luar. Aktivitas yang dilakukan di media sosial seperti memberi komentar, postingan dan ubah status dapat mengungkapkan informasi pribadi yang dapat dimanfaatkan. Teks yang ditinggalkan oleh pengguna tersebut dapat dianalisis untuk mendapatkan informasi yang diinginkan.

Pada jurnal ini akan dibahas tentang bagaimana membangun sistem untuk memprediksikan kepribadian melalui teks yang dituliskan oleh pengguna media sosial Twitter. Metode Naive Bayes, K-Nearest Neighbors, dan Support Vector Machine digunakan untuk mengklasifikasikan teks kepada jenis kepribadian yang ada.

II. TINJAUAN PUSTAKA

A. Big Five Personality

Personality atau kepribadian adalah kombinasi dari sifat dan tingkah laku seseorang dalam menghadapi berbagai situasi. Kepribadian dibagi menjadi 5 kategori utama yang disebut Big Five Personality Model [4] yaitu:

- *Agreeableness* atau keramahan, berkaitan dengan fokus seseorang memelihara hubungan sosial dengan orang lain. *Agreeableness* tinggi cenderung mempercayai orang lain dan dapat berkompromi.
- *Conscientiousness* atau sifat kehati-hatian, berhubungan dengan organisasi kehidupan seseorang. Individu dengan *Conscientiousness* tinggi biasanya hidup teratur, dapat diandalkan dan konsisten. Sebaliknya individu dengan nilai rendah biasanya santai, spontan, kreatif dan toleran.
- *Extraversion* mengukur kecenderungan seseorang untuk mencari hubungan dan mengekspresikan dirinya terhadap orang lain. Seorang *Extrovert* cenderung ramah, aktif, energetik dan suka berbicara. Sementara kebalikannya, *Introvert* lebih menghindari hubungan dengan orang lain.
- *Neuroticism* mengukur tingkat tendensi perubahan *mood*/suasana hati seseorang. Orang dengan *Neuroticism* tinggi berarti lebih mudah mengalami perubahan suasana hati dan terpengaruh dengan emosi negatif seperti *stress* atau gugup.
- *Openness* atau keterbukaan, berkaitan dengan imajinasi, kreativitas dan keingintahuan. Orang yang memiliki *Openness* tinggi memiliki sifat menyukai

perubahan, mengapresiasi sesuatu yang baru dan tidak biasa.

Big Five Personality adalah model kepribadian yang paling banyak diteliti pada bidang psikologi. Big Five Personality menunjukkan konsistensi pada wawancara, deskripsi diri dan observasi. Selain itu, Big Five Personality juga konsisten ditemukan dalam berbagai usia dan budaya yang berbeda. Pengukuran Big Five Personality banyak dilakukan melalui pengisian deskripsi diri maupun kuesioner [2].

B. Penelitian Terkait

Penelitian [3] melakukan percobaan prediksi kepribadian secara otomatis menggunakan fitur linguistik dari teks tertulis dan percakapan. Fitur linguistik yang digunakan adalah kata berbahasa Inggris berdasar aplikasi LIWC. Model kepribadian yang digunakan adalah Big Five Personality.

Penelitian [5] melakukan percobaan pencarian kepribadian dari fitur yang ditemukan dari Facebook. Fitur yang dicari linguistik dengan kata berbahasa Inggris berdasar aplikasi LIWC, struktural jaringan, aktivitas yang dilakukan dan informasi personal lainnya. Analisis menggunakan aplikasi Weka dengan 2 algoritma, M5 Rules dan Gaussian Processes.

Penelitian [6] menggunakan metode Naive Bayes untuk menentukan kepribadian dari sebuah teks yang dituliskan oleh seseorang. Tulisan yang dibuat adalah deskripsi diri sendiri yang akan digunakan untuk mencari kepribadian dan kemudian dicari pasangan pada situs pencarian jodoh. Bahasa yang digunakan adalah bahasa Indonesia. Model kepribadian yang digunakan adalah Four Temperaments yaitu *Sanguine, Choleric, Melancholic,* dan *Phlegmatic*.

C. Klasifikasi Teks

Klasifikasi teks berarti menentukan suatu dokumen berupa teks ke dalam suatu kelas atau kategori [7]. Sebelum memulai proses klasifikasi, data berupa teks harus terlebih dahulu diolah (*preprocessing*). Langkah *preprocessing* meliputi tokenisasi atau memecah kalimat menjadi kumpulan kata, *Stemming* atau proses yang mentransformasikan kata-kata yang terdapat pada suatu dokumen menjadi kata dasarnya (*root word*) dan pembobotan (*weighting*) yang dilakukan untuk membantu perhitungan. Salah satu metode pembobotan pada klasifikasi teks adalah TF-IDF. *Term Frequency* (TF) adalah jumlah kemunculan kata pada suatu dokumen. *Document frequency* (DF) adalah jumlah dokumen dimana terdapat kata tersebut. Perhitungan bobot TF-IDF ditunjukkan pada (1).

$$tfidf_t = f_{t,d} \times log \frac{N}{df_t} \quad (1)$$

Keterangan:
$tfidf_t$ = bobot kata *t*
$f_{t,d}$ = jumlah kemunculan kata t pada dokumen *d*
N = jumlah total dokumen
df_t = jumlah dokumen dimana terdapat kata *t*

D. Naive Bayes

Naive Bayes adalah metode klasifikasi data berdasarkan penerapan teorema probabilitas Bayes [8]. Multinomial Naive Bayes (MNB) adalah variasi dari Naive Bayes yang didesain untuk menyelesaikan permasalahan klasifikasi dokumen teks. MNB memanfaatkan distribusi multinomial dengan jumlah kemunculan kata atau bobot kata sebagai fitur klasifikasi. Persamaan MNB ditunjukkan pada (2).

$$c_{map} = \underset{c \in C}{\arg\max} \left[\log \frac{N_c}{N} + \sum_{1 \le k \le n_d} \log \frac{T_{ct} + \propto}{\sum_{t' \in V} (T_{ct'} + \propto)}\right] \quad (2)$$

Keterangan:
c_{map} = kelas pilihan
C = kumpulan semua kelas
n_d = jumlah kata pada dokumen *d*
N_c = jumlah dokumen pada kelas *c*
N = jumlah total dokumen
$P(t|c)$ = probabilitas kata *t* pada kelas *c*
T_{ct} = total bobot kata *t* pada dokumen latih di kelas *c*
V = *vocabulary* atau kumpulan kosa kata
$\propto$ = nilai *smoothing*

Pada perhitungan probabilitas kata terdapat parameter $\propto$ atau disebut juga *smoothing* yaitu penambahan suatu nilai pada setiap kata. *Smoothing* dilakukan untuk menghindari perhitungan nilai nol jika suatu kata tidak terdapat kemunculannya pada suatu dokumen. Nilai $\propto$ merupakan parameter yang dapat diatur dan memengaruhi probabiltas keluaran dari metode MNB.

E. K-Nearest Neighbors

K-Nearest Neighbors (KNN) adalah algoritma klasifikasi yang menggunakan fungsi jarak antara data percobaan dengan data pelatihan serta jumlah tetangga terdekat untuk menentukan hasil klasifikasi [8]. Fungsi jarak yang digunakan adalah *cosine similarity*. *Cosine similarity* adalah salah satu fungsi yang banyak digunakan dalam klasifikasi dokumen untuk menentukan kesamaan antara beberapa dokumen. Jarak yang dekat menujukkan kesamaan antara 2 dokumen sehingga memiliki kategori yang sama. Persamaan penilaian skor KNN ditunjukkan pada (3).

$$score(c, d_1) = \sum_{d_2 \in S_k d_1} I_c(d_2) \cos\ (vd_2, vd_1) \quad (3)$$

Keterangan:
$score(c, d_1)$ = nilai skor dokumen uji pada kelas *c*
d_1 = dokumen uji
d_2 = dokumen latih
vd_1 = vektor dokumen uji
vd_2 = vektor dokumen latih
I_c = 1 jika d_2 adalah anggota kelas *c*, 0 jika tidak
$S_k d_1$ = kumpulan dokumen *k*-terdekat dari dokumen uji

Penentuan kelas dilakukan dengan *voting* pada *K* tetangga yang terdekat. Tetangga terdekat merupakan *K* dokumen dengan nilai *similarity* tertinggi. Penentuan nilai *K* sangat krusial. Nilai *K* yang kecil berarti *noise* akan memiliki pengaruh lebih besar pada saat dilakukan *voting*. Menambah nilai *K* yang besar akan menambah juga jumlah *noise* namun pengaruhnya dapat berkurang sehingga ketepatan akan dapat meningkat.

F. Support Vector Machine

Support Vector Machine (SVM) adalah metode *supervised learning* yang menganalisis data dan mengenali pola yang digunakan untuk klasifikasi [8]. SVM mengambil himpunan pelatihan data dan menandai sebagai bagian dari suatu kategori kemudian memprediksi suatu masukan apakah merupakan anggota dari kelas yang ada. Model SVM merepresentasi data sebagai titik dalam ruang, dipetakan sehingga terpisah berdasar kategori yang dibagi oleh *hyperplane*/garis pemisah. Fungsi pencarian *hyperplane* optimum ditunjukkan pada (4) yang memenuhi (5).

$$\frac{1}{2}w^T w + C\sum_i \xi_i \tag{4}$$

$$\{(x_i, y_i)\}, y_i(w^T x_i + b) \geq 1 - \xi_i \tag{5}$$

Keterangan:
w = *weight vector*
C = fungsi *loss*
ξ_i = variabel *slack*/kesalahan klasifikasi vektor ke-*i*
x_i = *train vector* ke-*i*
y_i = kelas label *train vector* ke-*i*
b = nilai *bias*

Nilai *C* adalah nilai fungsi *loss* yaitu seberapa besar keinginan menghindari kesalahan klasifikasi. Semakin besar nilai *C*, semakin besar pula toleransi jarak dari *hyperplane*. Nilai *C* merupakan parameter yang dapat diatur dan memengaruhi probabilitas keluaran dari metode SVM.

III. ANALISIS DAN PERANCANGAN SISTEM

A. Persiapan Data

Data pelatihan berasal myPersonality Project [9]. myPersonality Project adalah aplikasi Facebook yang digunakan untuk test kepribadian berdasar pengisian kuesioner secara online. *Dataset* berupa 10.000 *status updates* dari 250 orang. Data teks yang didapat diproses terlebih dahulu. Semua post dari satu *userID* digabungkan/*append* menjadi satu baris *string* panjang yang dianggap sebagai satu dokumen. Dilakukan penghilangan 10 orang karena hanya terdapat 1-5 kata pada teks yang ditulisnya. *Dataset* akhir berupa 240 dokumen teks untuk tiap orang yang sudah dilabelkan. Pengklasifikasian bahasa Indonesia menggunakan *dataset* yang sama dengan menerjemahkan seluruh isi menjadi bahasa Indonesia. Asumsi yang digunakan adalah arti untuk setiap kata tetap akurat walaupun telah dialih-artikan ke bahasa lain. Cara ini memiliki keterbatasan yaitu adanya kemungkinan kesalahan dalam pengartian kata yang bermakna ganda atau ambigu, kata yang tidak ada padanannya dalam bahasa Indonesia dan kata yang memiliki perbedaan makna mengikuti konteks kalimat.

Data yang akan diprediksi berasal hasil pengambilan teks *tweet* dari seorang pengguna Twitter. Data teks yang diambil berupa 1.000 *tweet* terakhir pengguna baik berupa *tweet* yang dituliskan langsung oleh sang pengguna tersebut maupun data teks berupa *retweet* (tulisan pengguna lain yang disebarkan ulang). Kumpulan *tweet* dari pengguna juga dijadikan menjadi sebuah dokumen/*string* panjang.

B. Pengolahan Data

Data teks direpresentasikan menjadi model *Bag of Words* yaitu memecah kalimat menjadi kata-kata yang menyusunnya. Langkah-langkah yang dilakukan pada pengolahan data adalah:

- Tokenisasi yaitu mengubah kalimat menjadi kumpulan kata-kata tunggal yang menyusunnya.
- *Stemming* yaitu mengembalikan sebuah kata menjadi bentuk dasar dengan menghilangkan imbuhan yang ada. Algoritma stemming untuk bahasa Inggris yang digunakan adalah Porter Stemmer. Sementara algoritma untuk bahasa Indonesia adalah algoritma Nazief-Andriani [10].
- Menghilangkan *stop words*. *Stop words* adalah kata-kata yang tidak atau sedikit memiliki arti namun diperlukan dalam struktur gramatikal suatu bahasa [7].
- Pembobotan menggunakan metode TF-IDF untuk setiap kata.

Pada *dataset* myPersonality terdapat ±10.000 kata berbeda Dalam pencarian kata untuk dijadikan fitur klasifikasi dibatasi dengan jumlah 750 kata yang paling sering muncul dalam *dataset*. Pembatasan jumlah kata dilakukan untuk mengurangi beban dan waktu proses, menambah efektifitas dan meningkatkan akurasi.

C. Klasifikasi Data

Permasalahan yang diangkat merupakan permasalahan klasifikasi multi-label, yaitu satu orang dapat memiliki satu atau lebih ciri kepribadian atau bahkan tidak memiliki ciri kepribadian sama sekali, maka digunakan metode *multilabel classification*. Metode multi-label yang digunakan adalah *binary relevance* yaitu mentransformasikan label secara biner untuk setiap label terhadap label lainnya dengan asumsi independen atau sering disebut *One vs Rest* [11]. Penyelesaian permasalahan adalah dengan membuat *classifier* sebanyak jumlah label yang ada dan dilatih berdasar data yang sudah ditransformasikan. Setiap *classifier* merupakan *binary classifier* yaitu akan memberi keluaran apakah dokumen percobaan merupakan anggota pada label tersebut atau tidak.

Metode klasifikasi yang digunakan adalah Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN) dan Support Vector Machine (SVM). Untuk optimasi hasil klasifikasi, dilakukan pengaturan parameter yang dapat diubah pada setiap metode. Tabel 1 merupakan daftar parameter yang dapat diubah pada masing-masing metode beserta nilai parameter yang digunakan.

Tabel 1.
Daftar parameter setiap metode

Metode	Parameter	Nilai optimal
MNB	$\propto$ / nilai *smoothing*	1
KNN	K / jumlah *neighbors*	27
SVM	C / fungsi *loss*	1

Untuk mengatasi distribusi yang tidak seimbang antara jumlah orang yang memiliki suatu kepribadian dan tidak pada *dataset*, dilakukan pengaturan *threshold* untuk keputusan setiap *classifier* dalam menentukan data yang diuji apakah termasuk dalam label kepribadian tersebut. Penentuan titik *threshold* diambil dari nilai *F-Score* terbesar dari suatu titik keputusan *classifier*. *F-Score* (8) merupakan nilai *mean* dari *True Positive Rate,* ditunjukkan pada (6) dan *True Negative Rate* (7).

$$TPR = \frac{TP}{TP+FP} \quad (6)$$

$$TNR = \frac{TN}{FP+TN} \quad (7)$$

$$\text{F-Score} = 2 \times \frac{TPR \times TNR}{TPR+TNR} \quad (8)$$

Keterangan:
TPR = *True Positive Rate/Sensitivity*
TNR = *True Negative Rate/Specificity*
TP = *True Positive*
TN = *True Negative*
FP = *False Positive*
FN = *False Negative*
F-Score = nilai *F-Score*

Nilai *threshold* optimal yang didapat untuk setiap *classifier* dari setiap metode klasifikasi pada bahasa Inggris ditunjukkan pada Tabel 2. Nilai *threshold* optimal yang didapat untuk setiap *classifier* dari setiap metode klasifikasi pada bahasa Indonesia ditunjukkan pada Tabel 3.

Setiap metode mengeluarkan hasil prediksi yang dapat berbeda antara satu dengan yang lainnya. Untuk menghindari kebingungan dari hasil kesimpulan, dibuat satu hasil keluaran berupa prediksi gabungan yang diambil dari ketiga metode yang ada. Hasil gabungan merupakan digunakan *majority vote* dari ketiga metode. Sebagai contoh jika pada suatu label terdapat 2 atau lebih metode yang mengeluarkan prediksi ”Ya” maka hasil prediksi akhir label tersebut adalah ”Ya”. Sebaliknya jika lebih dari 2 metode menghasilkan ”Tidak” maka hasil prediksi gabungan adalah ”Tidak”.

IV. UJI COBA DAN EVALUASI

A. Pengujian Metode

Pengujian dilakukan dengan metode 10 *cross-validation* terhadap *dataset* myPersonality berbahasa Inggris kepada masing-masing label dari setiap metode. *Cross-validation* dilakukan sebanyak 10 kali dengan pembagian 90% data pelatihan dan 10% data percobaan. Hasil pengujian *dataset* bahasa Inggris ditunjukkan pada Tabel 4. Hasil pengujian *dataset* bahasa Inggris ditunjukkan pada Tabel 5.

Tabel 2.
Nilai *threshold* untuk setiap *classifier dataset* bahasa Inggris

Label	MNB	KNN	SVM
AGR	0,58	0,56	0,25
CON	0,48	0,4	-0,15
EXT	0,34	0,45	-0,1
NEU	0,32	0,33	-0,1
OPN	0,77	0,74	0,25

Tabel 3.
Nilai *threshold* untuk *classifier dataset* bahasa Indonesia

Label	MNB	KNN	SVM
AGR	0,59	0,59	0,15
CON	0,49	0,44	-0,1
EXT	0,35	0,41	-0,05
NEU	0,3	0,33	-0,05
OPN	0,76	0,74	0,3

Keterangan:
AGR = Label *Agreeableness*
CON = Label *Conscientiousness*
EXT = Label *Extraversion*
NEU = Label *Neuroticism*
OPN = Label *Openness*

Tabel 4.
Akurasi percobaan *dataset* bahasa Inggris

Akurasi	MNB	KNN	SVM
AGR	61,33 ± 1,89	57,38 ± 1,67	61,88 ± 1.97
CON	62,87 ± 1,54	63,00 ± 1,68	60.75 ± 1,16
EXT	57,75 ± 1,26	61,46 ± 2,11	59,99 ± 1.75
NEU	57,54 ± 1,9	52,25 ± 1,86	57,30 ± 1.29
OPN	63,67 ±1,54	64,92 ± 2,49	59.62 ± 1.54
Rata-rata	60,63 ± 1,64	59,8 ± 1,98	59,62 ± 1,54

Tabel 5.
Akurasi percobaan *dataset* bahasa Indonesia

Akurasi	MNB	KNN	SVM
AGR	63,21 ± 1,84	60,71 ± 2,48	62,95 ± 1,2
CON	58,67 ± 2,1	61,08 ± 3,19	60,35 ± 1,38
EXT	51,5 ± 2,02	52,87 ± 2,03	53.21 ± 2,17
NEU	54,63 ± 2,01	54,0 ± 1,83	54,2 ± 2,14
OPN	72,29 ± 2,13	62,83 ± 2,07	65,75 ± 1.69
Rata-rata	60,06 ± 2,02	58,30 ± 2,37	59,29 ± 1,76

Tabel 6.
Akurasi percobaan terhadap 40 responden

Metode	Akurasi
MNB	63 %
KNN	60 %
SVM	61 %
Gabungan	65 %

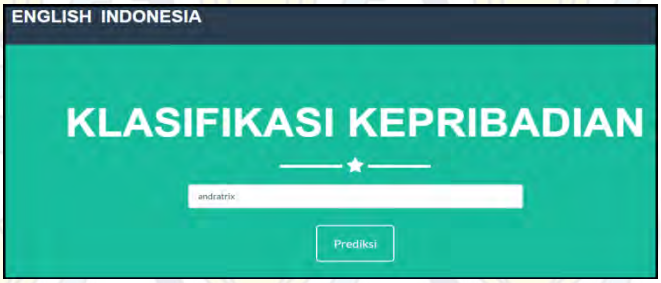
Skenario selanjutnya adalah dilakukan pengambilan data teks dari akun Twitter dari seorang pengguna. Teks kemudian diklasifikasikan menggunakan 3 metode yang ada dan metode gabungan yang merupakan hasil akhir dari aplikasi. Hasil prediksi kemudian diadu dengan hasil prediksi kepribadian dari pengisian kuesioner IPIP 50-Item Set of IPIP Big-Five Factor Markers [12]. Responden terdiri dari 40 orang. Responden yang dipilih harus memiliki akun Twitter dengan jumlah *tweet* minimal sebanyak 1.000. Responden akan mengisi kuesioner dan melaporkan hasilnya. Sementara sistem akan mengambil teks dari akun Twitter responden untuk diklasifikasikan. Hasil pengujian responden ditunjukkan pada Tabel 6.

Dari 3 skenario percobaan yang dilakukan, metode MNB konsisten menjadi yang terbaik meski dengan selisih yang sedikit dibandingkan metode lain. Pada pengujian *cross-validation*, MNB mendapatkan nilai akurasi terbaik dari ketiga metode yang digunakan. Pada pengujian responden metode

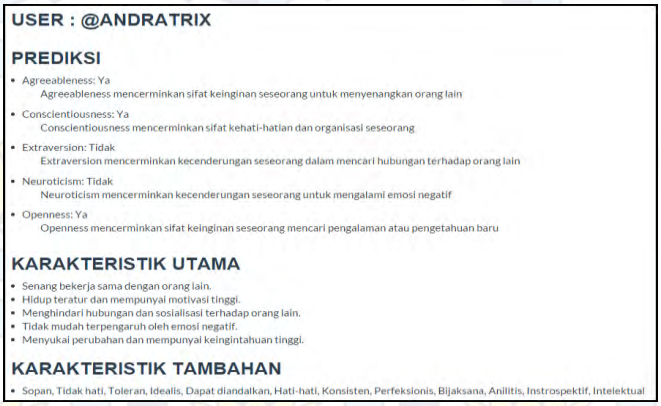
MNB juga mendapatkan hasil terbaik kedua setelah metode gabungan. Metode SVM secara keseluruhan sedikit dibawah MNB dikarenakan kesulitan memisahkan kelas suatu kata dalam *dataset* yang kurang akurat. Metode KNN merupakan metode dengan tingkat akurasi terendah. Dugaan penyebab rendahnya nilai metode KNN karena kesulitan dalam penentuan nilai *K* yang optimal. Jumlah nilai *K* sangat menentukan hasil akhir karena akan dilakukan perhitungan probabilitas pada *K* sampel. Ini berbeda dengan MNB dan SVM yang menggunakan perhitungan murni pada fitur yang ada. Sementara itu metode gabungan yang merupakan prediksi akhir dari aplikasi mendapatkan hasil terbaik pada pengujian responden yaitu 65%. Metode gabungan dapat menghasilkan akurasi yang lebih baik karena terdapat perbaikan klasifikasi pada hasil. Pada kasus dimana salah satu metode kurang tepat dalam melakukan klasifikasi, maka akan dapat tertutupi oleh kedua metode lainnya jika prediksi dari metode lain tersebut adalah benar.

B. Pengujian Fungsionalitas

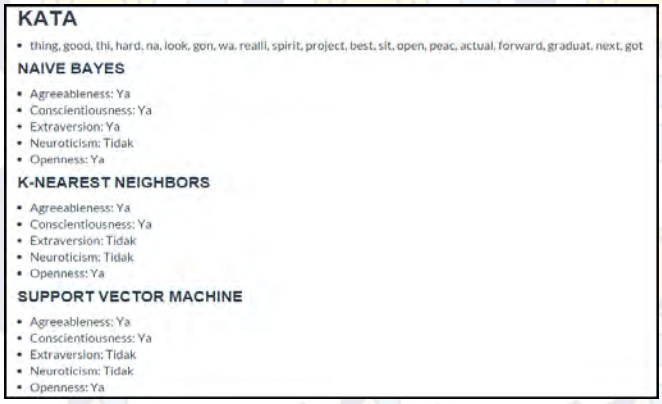
Pengujian fungsionalitas dilakukan dengan menjalankan aplikasi dan menentukan apakah aplikasi telah sesuai dengan perancangan. Gambar 1 merupakan tampilan halaman utama dimana pengguna dapat memasukkan nama akun Twitter. Gambar 2 merupakan hasil akhir keluaran aplikasi berupa prediksi terhadap 5 tipe kepribadian. Selain itu juga dihasilkan karakteristik kepribadian utama yang didapatkan dari sifat kelima label kepribadian dan juga karakteristik kepribadian sekunder yang didapatkan dari kombinasi antar label kepribadian. Gambar 3 adalah detil prediksi dari masing-masing metode. Dapat disimpulkan bahwa aplikasi berjalan dengan baik dan memenuhi kebutuhan keluaran yang diinginkan.



Gambar 1. Halaman utama aplikasi



Gambar 2. Hasil prediksi kepribadian seorang pengguna



Gambar 3. Detil prediksi untuk masing-masing metode

V. KESIMPULAN

Prediksi kepribadian berdasarkan teks yang ditulis dari Twitter berhasil dilakukan dengan melakukan klasifikasi teks yang didapat secara *supervised learning* atau sudah dilabelkan sebelumnya. Dari tiga metode yang digunakan, Naive Bayes sedikit lebih unggul dari metode lainnya.

Saran yang dapat diberikan adalah penggunaan *dataset* yang lebih akurat untuk meningkatkan akurasi dan penggunaan *dataset* bahasa Indonesia yang diambil langsung dari teks yang berbahasa asli Indonesia, bukan diterjemahkan. Selain itu dapat dikembangkan pula pendekatan secara semantik dengan mencari makna dari suatu kata.

DAFTAR PUSTAKA

[1] Cantandir, I., Fernandez-Tobiaz, I., Belllogin, A., "Relating personality types with user preferences in multiple entertainment domains," EMPIRE 1st Workshop on Emotions and Personality in Personalized Services, 2013.

[2] Goldberg, L.R., Johnson, J.A., Eber, H.W., Hogan, R., Ashton, M.C., Cloninger, C.R., "The International personality item pool and the future of public domain personality measures," Journal of Research in Personality, 40(1), 84-96, 2006.

[3] Mairesse, F., Walker, M., Mehl, M., Moore, R., "Using linguistic cues for the automatic recognition of personality in conversation and text," Journal of Artificial Intelligence Research (JAIR). 30(1), 457–500, 2007.

[4] Costa, P.T., McCrae, R.R., "Revised NEO personality inventory (NEO-PI-R) and NEO five-factor inventory (NEO-FFI)," Psychological Assessment Resources, 1992.

[5] Golbeck, J., Robles C., Turner, K., "Predicting personality with social media," In CHI'11 Extended Abstracts on Human Factors in Computing Systems, pp. 253-262, 2011.

[6] Lestari, N.M.A., Putra, I.K.G.D., Cahyawan, A.A.K.A., "Personality types classification for Indonesian text in partners searching website using Naive Bayes methods," International Journal of Software and Informatics Issues, 2013.

[7] Garnes, Ø. L. *Feature Selection for Text Categorization*. Oslo, 2009.

[8] Manning, C.D., Raghavan, P., Schutze, H. *Introduction to Information Retrieval*. Cambridge UP, 2008.

[9] Celli, F., Pianesi, F., Stillwell, D., Kosinski, M., "Workshop on computational personality recognition (shared task)," In Proceedings of WCPR13, in conjunction with ICWSM-13, 2013.

[10] Nazief, B.A.A., Adriani, M., "Confix-stripping: approach to stemming algorithm for Bahasa Indonesia," Internal publication, Faculty of Computer Science, Univ. of Indonesia, Depok, Jakarta, 1996.

[11] Tsoumakas, G., Katakis, I., "Multi-label classification: an overview," International Journal of Data Warehousing & Mining 3 (3): 1–13, 2007.

[12] Goldberg, Lewis R., "The Development of markers for the Big-Five Factor structure," Psychological Assessment 4.1: 26, 1992.

BIODATA PENULIS

Bayu Yudha Pratama lahir di kota Bojonegoro pada tanggal 17 April 1993. Penulis adalah anak pertama dari dua bersaudara dan besar di kota Surabaya.

Penulis menempuh pendidikan formal di SD Batan Indah Tangerang (1999-2001), SDN Randu Agung II Gresik (2002-2003), SDN Sememi I Surabaya (2004-2005), SMPN 1 Surabaya (2005-2008), SMAN 2 Surabaya (2008-2011). Saat ini penulis menempuh pendidikan di S1 Jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur dengan NRP 5111100127. Di jurusan Teknik Informatika, penulis mengambil bidang minat Manajemen Informasi dan memiliki ketertarikan di bidang *machine learning* dan *text mining*. Penulis dapat dihubungi melalui alamat email: bayuyp@gmail.com.

Klasifikasi Kepribadian Berdasarkan Tulisan dari Twitter Menggunakan Metode Naive Bayes, KNN dan SVM

Bayu Yudha Pratama
5111100127

Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D
Ratih Nur Esti A S.Kom, M.Sc

Latar Belakang

Mengapa Kepribadian?

- Perekrutan pegawai
- Konseling karir
- Konseling relationship
- Konseling kesehatan
- Pemilihan produk (film, musik, buku, politik, dll)

Latar Belakang

Pengukuran Kepribadian?

- Wawancara psikolog
- Observasi psikolog
- Pengisian kuesioner

Masalah?

- Tidak praktis
- Mahal
- Memerlukan waktu

Latar Belakang

Cara lain?

- Ciri kepribadian seseorang dapat diperoleh secara otomatis dari teks yang dituliskannya.
- Pemilihan kata-kata yang sering digunakan dapat menggambarkan kepribadian orang tersebut.

Latar Belakang

Mengapa sosial media?

- Bersifat pribadi.
- Merepresentasikan diri seseorang.
- Tulisan dapat diakses oleh umum.

Latar Belakang

Permasalahan

- Bagaimana mendapatkan data teks dari pengguna Twitter?
- Bagaimana mengklasifikasikan data teks sesuai dengan model kepribadian?
- Bagaimana membandingkan hasil yang diperoleh dari metode Naive Bayes, Support Vector Machine dan k-Nearest Neigbors?

Latar Belakang

Batasan masalah

- Menggunakan metode supervised learning atau sudah dilabelkan sebelumnya.
- Kumpulan kosa kata yang digunakan adalah bahasa Indonesia dan Inggris.
- Media sosial yang digunakan adalah Twitter.

Latar Belakang

Tujuan

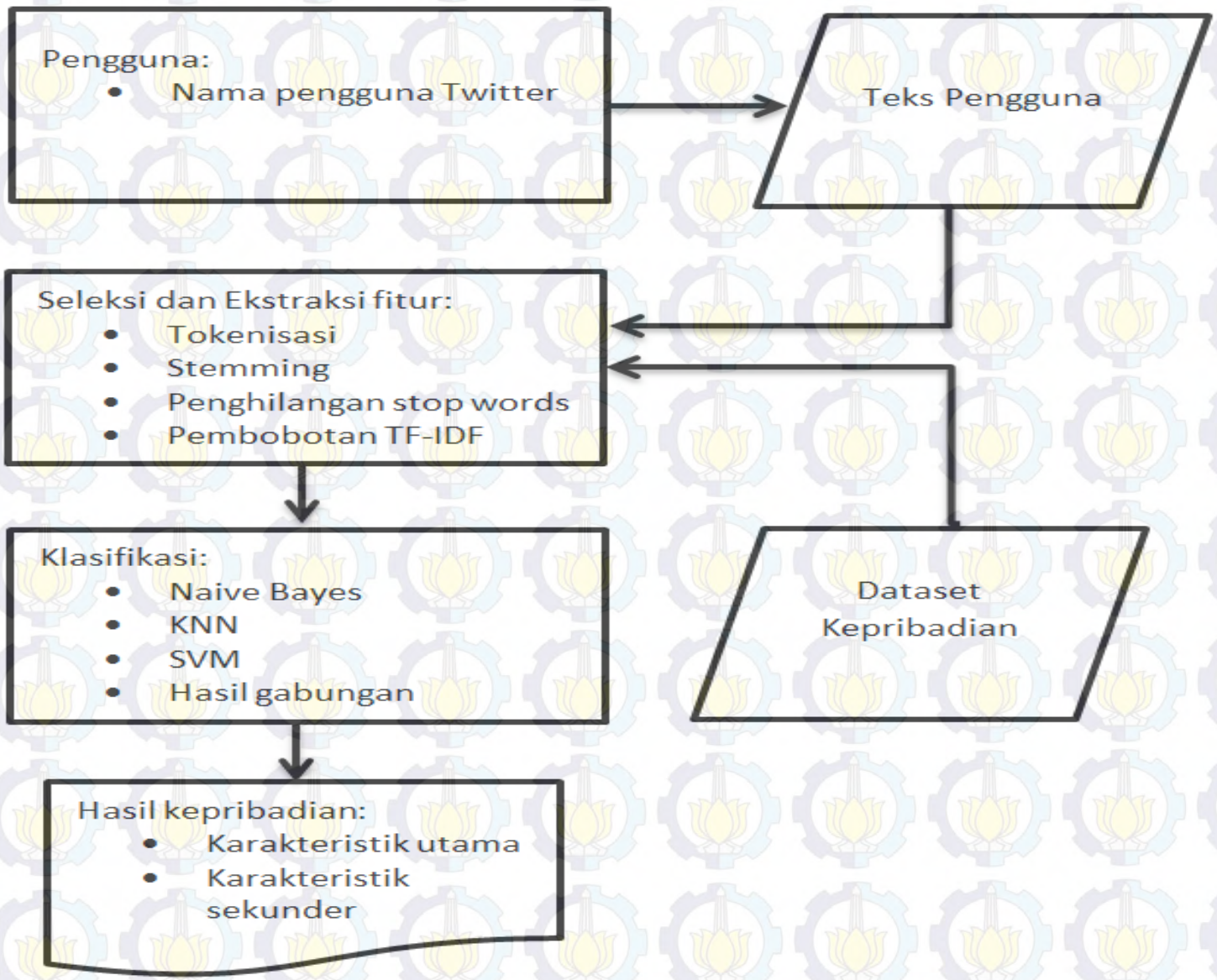
- Memprediksi kepribadian pengguna’
- Membandingkan hasil metode

Latar Belakang

Tinjauan Pustaka

- Big Five Personality model
- Text processing
- Multilabel classification
- Naive Bayes
- K-Nearest Neighbors
- Support Vector Machine

Flowchart


Pengguna:
Nama pengguna Twitter
Teks Pengguna
Seleksi dan Ekstraksi fitur:
Tokenisasi
Stemming
Penghilangan stop words
Pembobotan TF-IDF
Klasifikasi:
Naive Bayes
KNN
SVM
Hasil gabungan
Dataset Kepribadian
Hasil kepribadian:
Karakteristik utama
Karakteristik sekunder

Big Five Personality Model

Big Five Personality model membagi kepribadian menjadi 5 buah ciri utama:

- Agreeableness, menunjukkan nilai individu saat bekerja sama atau bersosialisasi dengan orang lain
- Conscientiousness, menunjukkan kedisiplinan diri, kepatuhan dan keinginan untuk mendapat pencapaian.
- Extraversion, menunjukkan kecenderungan untuk bersosialisasi dan berinteraksi kepada orang lain

Big Five Personality Model

Big Five Personality model membagi kepribadian menjadi 5 buah ciri utama:

- Neuroticism, menunjukkan kecenderungan untuk terpengaruhi oleh emosi negatif seperti gugup, marah, depresi
- Opennes, menunjukkan kecenderungan apresiasi terhadap ide, imajinasi, seni dan keingintahuan.

Text Processing

Sebelum dilakukan klasifikasi, data teks harus diproses terlebih dahulu.

- Tokenisasi, memecah kalimat menjadi kumpulan kata-kata tunggal yang menyusunnya.
- Stemming, mengubah kata menjadi bentuk dasanya.
- Penghilangan stop words atau kata-kata yang tidak mempengaruhi proses klasifikasi.

Text Processing

- Pembobotan, untuk setiap kata dilakukan pembobotan untuk perhitungan klasifikasi. Pembobotan menggunakan metode TF-IDF.

$$tfidf_t = f_{t,d} \; x \; log \frac{N}{df_t}$$

Multilabel Classification

- Karena topik adalah multi-label, yaitu satu orang dapat lebih mempunyai lebih dari 1 kepribadian, maka digunakan metode multilabel classification.
- Metode multi-label yang digunakan adalah binary relevance yaitu mentransformasikan label secara binary untuk setiap label terhadap label lainnya(One vs Rest).

Multilabel Classification

- Setiap classifier akan memberi keluaran apakah dokumen tes masuk pada label tersebut (Yes/No).

Dataset

	AGR	CON	EXT	NEU	OPN
U1	Y	Y	N	N	Y
U2	Y	N	Y	Y	N
U3	N	N	Y	N	Y
U4	Y	Y	N	N	Y
U5	N	Y	Y	N	N

Transformed Dataset

U1	1	U1	1	U1	0	U1	0	U1	1
U2	1	U2	0	U2	1	U2	1	U2	0
U3	0	U3	0	U3	1	U3	0	U3	1
U4	1	U4	1	U4	0	U4	0	U4	1
U5	0	U5	1	U5	1	U5	0	U5	0

AGR CLASSIFIER | CON CLASSIFIER | EXT CLASSIFIER | NEU CLASSIFIER | OPN CLASSIFIER

Naive Bayes

- Naive Bayes adalah metode klasifikasi yang menggunakan perhitungan probabilitas.

$$C_{map} = \arg\max \left[\log P\,(c) + \sum_{i=1}^{n} \log P\,(t|c) \right]$$

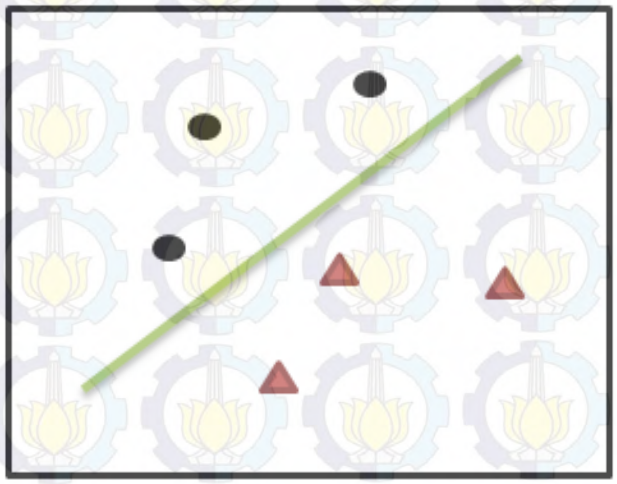
K-Nearest Neighbors

- KNN adalah metode klasifikasi yang menggunakan fungsi jarak antara data pelatihan dan data percobaan.
- Fungsi yang digunakan adalah cosine similarity.

$$sim(d1, d2) = \frac{d1 \circ d2}{||d1||||d2||}$$
$$= \frac{\sum_{i=1}^{n} d1\ x\ d2}{\sqrt{\sum_{i=1}^{n}(d1)^2}\ x\ \sqrt{\sum_{i=1}^{n}(d2)^2}}$$

Support Vector Machine

- SVM adalah metode klasifikasi yang memprediksikan hasil dengan membagi data pelatihan sebagai ruang.



Dataset

- Dataset pelatihan yang digunakan berasal myPersonality Project.
- Dataset berupa 9900 status updates Facebook dari 250 orang pengguna.
- Setiap pengguna sudah dilabelkan berdasar Big Five Personality.

USER	TEXT	AGR	CON	EXT	NEU	OPN
user1	"...when will the GSS end? is BOSE headphone part of the GSS? w	y	n	n	n	n
user2	"Holy Haruhi, season 2's out?!" A recent online exchange: Other g	n	n	n	y	n
user3	"In my days, we played soccer in the minefield." wants rime with	n	n	y	n	n
user4	"Those who criticize our generation forget who raised it." "In awe	y	y	n	y	y
user5	"we never forget the truth, we only get better at lying to ourselv	n	y	y	n	y
user6	(purposefully contradicting *PROPNAME*) AGHH!! The heat! *op	y	y	n	n	y
user7	...YAWN... You no black eye. Boo no fencing tomorrow night. You	n	y	n	n	y

Dataset

- Klasifikasi bahasa Indonesia menggunakan datatest sama dengan menerjemahkan seluruh isi menjadi bahasa Indonesia.

USER	TEXT	AGR	CON	EXT	NEU	OPN
user1	akan Headphone wan membayar membenci kehidupan pek	y	n	n	n	n
user2	musim suci pertukaran pria membayangkan sepenuhnya me	n	n	n	y	n
user3	hari bermain sepakbola rime belajar minuman keras high he	n	n	y	n	n
user4	mengkritik generasi lupa mengangkat kagum menyaksikan v	y	y	n	y	y
user5	melupakan kebenaran yang lebih baik berbohong akan tidu	n	y	y	n	y
user6	sengaja bertentangan panas membuka lengan menyambut k	y	y	n	n	y
user7	menguap yay mata hitam boo pagar besok malam menguap	n	y	n	n	y
user8	gambar gelembung sepeda catatan yang diminta korban jatu	n	y	n	n	y

Data Percobaan

- Data yang akan diprediksi berasal hasil pengambilan teks tweet dari seorang pengguna Twitter.
- Kumpulan tweet dari pengguna juga dijadikan menjadi sebuah dokumen/string panjang.

Happy #OpeningDay. This would be huge for working families: #LeadOnLeave Today is the 137th White House Easter Egg Roll. Check out the celebration here: #GimmeFive WATCH: In the weekly address, President Obama talks about the historic understanding reached with Iran. "From my family to yours, Chag Sameach." —President Obama The President is launching an initiative to train 75,000 Americans—including veterans—to join the solar workforce: "We've got to lead by example, invest in the future, train our workers to get jobs in the clean energy economy." —President Obama "We've got to be relentless in our work to grow the economy and create new jobs." —President Obama "Since I took office, solar electricity has gone up twenty-fold." —President Obama #ActOnClimate LIVE: President Obama is in Utah talking about training American workers for clean energy jobs. #ActOnClimate Tune in at 1 p.m. ET to watch President Obama at Hill Air Force Base in Utah discuss the importance of clean energy: Our economy added 126,000 jobs in March, the 61st consecutive month of private-sector job growth. LIVE: President Obama is speaking in Louisville, Kentucky, about the economy. Tune in at 5:50 p.m. ET to watch the President deliver remarks on our economy in Louisville, Kentucky: We need a budget that works for every American—not just the wealthy few. Happening now: The President is delivering a statement on Iran. This 9-year-old girl stood up for what she believes in, and then got a letter back from the President: Read how #Obamacare is a major reason why we've seen an estimated 50,000 fewer preventable patient deaths: Renewable energy investments are up 17 percent globally from 2013. Read more: The budget resolutions Congress passed would devastate programs millions of middle-class families rely upon. Climate change is a global problem. Here's another step towards solving it: #ActOnClimate Get ready for the final sprint. You have until midnight to enter ... Go! This shouldn't be a debate. #ActOnClimate You could talk about anything—even compare jump shots. Enter now: Be one of the people building this grassroots movement from the ground up: Stay calm, but act fast: There's one day left to enter for the chance to meet President Obama. Another nation commits to #ActOnClimate—Mexico announced it would cut carbon pollution 25 percent by 2030: Free flight, free hotel, and a chance to meet the President? Enter today: LIVE: President Obama is speaking about the life and legacy of Senator Ted Kennedy in Boston. "Protecting working Americans' paychecks

Uji Coba

Skenario 1

- 5-Cross validation pada dataset berbahasa Inggris

ACC	NB	KNN	SVM
AGR	61.33 ± 1.89	57.38 ± 4.49	62.38 ± 2.23
CON	62.87 ± 1.54	63.00 ± 1.68	61.25 ± 2.05
EXT	57.75 ± 1.26	61.46 ± 4.44	60.46 ± 3.35
NEU	57.54 ± 1.9	52.25 ± 5.88	57.83 ± 1.43
OPN	63.67 ±1.54	64.92 ± 2.82	58.67 ± 5.42
AVG	**60.63 ± 1.64**	**59.80 ± 4.13**	**60.12 ± 3.22**

Uji Coba

Skenario 2

- 5-Cross validation pada dataset berbahasa Indonesia

ACC	NB	KNN	SVM
OPN	63.21 ± 1.84	60.71 ± 3.62	63.64 ±1.26
CON	58.67 ± 2.1	61.08 ± 4.08	61.05 ± 2.9
EXT	51.5 ± 2.02	52.87 ± 2.49	53.76 ± 3.23
AGR	54.63 ± 2.01	54.0 ± 1.95	54.79 ± 2.26
NEU	72.29 ± 2.13	62.83 ± 10.18	66.53 ± 6.31
AVG	**60.06** ± 2.02	**58.3** ± 5.36	**59.95** ± 3.62

Uji Coba

Skenario 3

- Pengujian dengan 40 responden yang diadu dengan hasil prediksi pengisian kuesioner.

	ACC
NB	63.00%
KNN	60.00%
SVM	62.00%
FIN	62.00%

Antarmuka aplikasi

Tampilan

Antarmuka aplikasi

Hasil prediksi

USER : @ANDRATRIX

PREDIKSI

- Agreeableness: Ya
  Agreeableness mencerminkan sifat keinginan seseorang untuk menyenangkan orang lain
- Conscientiousness: Ya
  Conscientiousness mencerminkan sifat kehati-hatian dan organisasi seseorang
- Extraversion: Tidak
  Extraversion mencerminkan kecenderungan seseorang dalam mencari hubungan terhadap orang lain
- Neuroticism: Tidak
  Neuroticism mencerminkan kecenderungan seseorang untuk mengalami emosi negatif
- Openness: Ya
  Openness mencerminkan sifat keinginan seseorang mencari pengalaman atau pengetahuan baru

KARAKTERISTIK UTAMA

- Senang bekerja sama dengan orang lain.
- Hidup teratur dan mempunyai motivasi tinggi.
- Menghindari hubungan dan sosialisasi terhadap orang lain.
- Tidak mudah terpengaruh oleh emosi negatif.
- Menyukai perubahan dan mempunyai keingintahuan tinggi.

KARAKTERISTIK TAMBAHAN

- Sopan, Tidak hati, Toleran, Idealis, Dapat diandalkan, Hati-hati, Konsisten, Perfeksionis, Bijaksana, Anilitis, Instrospektif, Intelektual

Antarmuka aplikasi

Hasil prediksi tiap metode

Metode

KATA

- thing na reall gon good hard look beauti haha sit enjoy thought iri dang okay almost graduat mayb best next

NAIVE BAYES

- Agreeableness: Tinggi
- Conscientiousness: Tinggi
- Extraversion: Tinggi
- Neuroticism: Rendah
- Openness: Tinggi

K-NEAREST NEIGHBORS

- Agreeableness: Tinggi
- Conscientiousness: Tinggi
- Extraversion: Rendah
- Neuroticism: Rendah
- Openness: Tinggi

SUPPORT VECTOR MACHINE

- Agreeableness: Tinggi
- Conscientiousness: Tinggi
- Extraversion: Rendah
- Neuroticism: Rendah
- Openness: Tinggi

Demonstrasi aplikasi

Kesimpulan

- Data teks dari pengguna Twitter berhasil didapatkan dengan memanfaatkan API yang tersedia.
- Klasifikasi teks berdasar kepribadian berhasil diprediksikan dengan mengadunya dengan dataset yang sudah dilabelkan.
- Hasil yang diperoleh dari 3 metode dapat dibandingkan berdasar akurasi, sensitivity dan specificity.

Kesimpulan

- Metode Naive Bayes mendapatkan akurasi yang sedikit lebih baik dari metode lainnya.
- Sistem dapat memprediksikan kepribadian pengguna dari teks Twitter dengan ketepatan 60%.

Saran

- Penggunaan dataset yang lebih akurat untuk meningkatkan tingkat akurasi.
- Penggunaan dataset bahasa Indonesia yang diambil langsung dari teks berbahasa asli Indonesia.
- Penambahan media sosial lain.